JUNE 1993

Assessing Trends in the System Building Process of Selected Vendors

INTERIM REPORT

YA5

INPUT

The Atrium at Glenpointe, 400 Frank W. Burr Boulevard, Teaneck, NJ 07666 201-801-0050

Interim Report

ASSESSING TRENDS IN THE SYSTEM BUILDING PROCESS OF SELECTED VENDORS

I. BACKGROUND AND METHODOLOGY

A. Background

Andersen Consulting is considering making changes to the way in which its software solutions are created and delivered. To assist in this process Andersen engaged INPUT to examine what other vendors are doing or planning to do in this area.

Andersen and INPUT staff met in Chicago to review the background of the study on May 6, 1993. Over the next week the vendors to be targeted and the issues to be addressed were finalized.

B. Methodology

An interview guide was developed and reviewed by Andersen. The guide was intended to be a departure point for understanding the other vendor's views and plans across the broad area of system building. A copy of the Interview Guide follows the end of this chapter.

The following vendors were targeted:

Borland
CAP Gemini*
Cambridge Technology*
D. Appleton (DACOM)*
DEC*
EDS
HP*
IBM*
Microsoft
Oracle*
Texas Instruments
Xerox (Parc)

The firms followed by an * have been interviewed and their results are contained in the report. The others are in process and will be contained in the expanded final report.

INPUT has already made in excess of 75 contacts at these firms to obtain information.

- In some cases, a single person is able to provide all or most of the information required. This is generally because the company has centralized the function and/or has it well-coordinated. Even in this category, it has often taken a number of tries and referrals to find the correct person.

- In other cases (about half so far), responsibilities for software building are diffused throughout the organization. Up to six interviews have been required in this type of organization

INPUT was not sure at the beginning of the study what kind of reception the study would receive. In fact, every company approached has been very interested in the subject and, equally important, quite willing to take part in the study. The interview guide itself has been a significant "selling point" in motivating vendors to take part. The only limitations have been finding the right person and accommodating their schedule.

Because of this elevated level of interest INPUT has not found it necessary so far to promise anonymity. In a few cases proprietary or sensitive information has not been reported here; but this kind of information has been infrequent and, in INPUT's opinion, has no effect on findings.

INPUT believes that being able to associate the findings with actual companies, rather than, "A", "B", "C", etc. will make this information more useful for Andersen. INPUT does request Andersen, however, that this information be treated with care and not widely publicized, at least without masking the identities of the vendors adequately.

Three of the remaining companies are scheduled in the next several days and INPUT expects to have all companies' interviews completed by early July.

C. Organization of This Report

The remainder of this report is organized as follows:

Chapter II: Summary Analyses - This will be INPUT's conclusions based on the full set of interviews; this will appear in the final report.

Chapter III: Summary Matrixes - This very concisely summarizes material under the 13 categories in the Interview Guide. For more detail, the full text in Chapter IV should be consulted.

Chapter IV: Summary by Category - This provides the full text for each vendor within each category.

Chapter V: Summary by Vendor - This is the same information as in the preceding chapter, but organized so as to give a profile of each vendor.

CHAPTER II:

Summary Analysis

[This will appear in the final report.]

CHAPTER III:

Summary Matrixes

1. CHANGES IN THE SOFTWARE BUILDING PROCESS (OVERALL)

Cambridge Technology
Techniques: Rapid Prototyping
Development Time: Speeded up 3 times
Cost Reduction: Reduced
Quality: Increased

Cap Gemini
Techniques: JAD, RAD, partner tools
Development Time: Reduced by 25%
Cost Reduction: Somewhat
Quality: Some increase

DACOM
Techniques: Process Modeling
Development Time: Probably decreased
Cost Reduction: Probably decreased
Quality: Much higher

DEC
Techniques: JAD, RAD
Development Time: Reduce by 33% (goal)
Cost Reduction: Somewhat
Quality: Some increase

HP
Techniques: Focus on client/server
Development Time: Reduce by 25%
Cost Reduction: Reduced by 20%
Quality: High priority

IBM
Techniques: Many
Development Time: Reduce by 30% (?)
Cost Reduction: Reduce by 25-30%
Quality: Important

Oracle
Techniques: Rapid prototyping
Development Time: Reduce by 30%
Cost Reduction: No
Quality: Zero defects (goal)

2. SOFT BENEFITS

Cambridge Technology	o	Better communications
	o	Test concepts of reengineered business
	o	Improve requirements specifications
Cap Gemini	o	Help understand how business processes will change
DACOM	o	Benefits accrue to underlying business
	o	Reduced cycle time, improved maintenance
	o	End users are part of team
DEC	o	Better communications
	o	Improved customer satisfaction
	o	Increased responsiveness to customer needs
HP	o	Improved customer satisfaction
	o	Reduced cycle time for business transactions
IBM	o	Improved communications
	o	Improved customer satisfaction
	o	Customer review of systems
Oracle	o	Improved communications
	o	Replacement of paper with electronic interactions

3. RELATIVE CONTRIBUTION OF TOOLS AND PROCESSES

Cambridge Technology	**Tools:** Important to the extent they are part of the process **Processes:** Rapid prototyping very important; parallel teams with a factory-like flavor are also important
Cap Gemini	**Tools:** Tools supplied by partners play an important secondary role **Processes:** Process is most important; some consideration to factory approach; some dependence on skilled consultants
DACOM	**Tools:** Tools are secondary **Processes:** Modeling process is key; significant dependence on skilled consultants
DEC	**Tools:** Moving away from priority placed on tools **Processes:** JAD and RAD expected to have more impact; software factories being investigated
HP	**Tools:** CASE and other tools secondary **Processes:** Focussing most attention on process
IBM	**Tools:** Direction unclear, but tools are important **Processes:** Direction unclear
Oracle	**Tools:** Important, symbiotic with process **Processes:** Important, symbiotic with tools

4. REUSABILITY

Cambridge Technology	o o	Not a focus In principle, their process of separating functions in implementation supports reusability
Cap Gemini	o	Exploring feasibility
DACOM	o o	Do not see theories for supporting reusability in commercial market Have seen little potential for reusability in processes which appear similar on the surface
DEC	o o	Experimenting both internally and with their VARs Have tools to support reusability
HP	o o	Very little being done in custom code More attention in product area
IBM	o o	Experimenting Being used in system software development internally
Oraclé	o o o	Reusability is an objective More opportunities in an "all Oracle" environment Need to develop standards

5. CLIENT/SERVER MODEL

Cambridge Technology	**Problems Seen:** Incomplete model **Vendor Direction:** Integrate with reengineering
Cap Gemini	**Problems Seen:** Incomplete model [INPUT note: CGA may not fully understand issue.] **Vendor Direction:** Part of systems definition
DACOM	**Problems Seen:** Very incomplete model **Vendor Direction:** Understanding business process is critical to avoid painful mistakes
DEC	**Problems Seen:** Incomplete model **Vendor Direction:** Address technical issues
HP	**Problems Seen:** Comfortable with overall model; still need to link redesigned processes to client/server model **Vendor Direction:** Continue improvements
IBM	**Problems Seen:** Incomplete model **Vendor Direction:** Addressing separating of processes and data
Oracle	**Problems Seen:** Incomplete model, especially in heterogeneous platforms **Vendor Direction:** See standards as assistance

6. SYSTEM BUILDING MODELS/APPROACHES

Cambridge Technology	o o	Use a single process model High degree of management control
Cap Gemini	o o	Primary and secondary models Medium-high degree of management control
DACOM	o o	Use a single process model High degree of management control
DEC	o o	Multiple process models; efforts are coordinated Medium degree of management control
HP	o o	Multiple process models; coordination is attempted Medium-low degree of management control
IBM	o o	Multiple process models, more being added Low degree of management control
Oracle	o o	Multiple process models Low degree of management control

7. SKILL NEEDS AND DISTRIBUTION

Cambridge Technology	o	Skills in short supply, distributed unevenly
	o	Continuing education
Cap Gemini	o	Skill bottlenecks
	o	Use outside staff
	o	Continuing education
DACOM	o	Lack of skills is biggest problem
	o	Forced to hire senior staff
	o	Internal training is largely on the job
DEC	o	Need new skills, not always available internally
	o	Continuing education
	o	Use outside staffing sources
HP	o	Technical skills are not a problem, given culture
	o	Project management biggest gap
IBM	o	Skills are in short supply; cutbacks have made situation more acute
	o	Continuing training
	o	Use outside contractors
Oracle	o	Skills not perceived as a problem due to culture

8. SYSTEM TESTING PROCESS

Cambridge Technology	o o	Tested at prototype stage Heavy user involvement
Cap Gemini	o o	Tested at early stage Users involved
DACOM	o	Testing done at model level
DEC	o	Want to incorporate testing as part of system building process
HP	o o	Significant changes due to client/server Testing automated
IBM	o o	Changes made in tools and systems review Users increasingly brought into process
Oracle	o o	Now an on-going process involving customer Testing becoming much more complex; still involves considerable manual testing

9. INNOVATIONS (SELF-ASSESSED)

Cambridge Technology	o	Methodology
Cap Gemini	o	Combination of their own experience and tools from other vendors
DACOM	o o	Emphasis on modeling of business processes Making customer part of team
DEC	o	Factory-like code development
HP	o	Don't see themselves as innovative per se
IBM	o	Various technical initiatives, largely with partners
Oracle	o	Methodologies for continuous refinement of systems

10. COST OF MAKING IMPROVEMENTS IN SYSTEM BUILDING

Cambridge Technology	o o	Sees payoff in near term Controls in place to balance cost versus benefits
Cap Gemini	o o	Seeks to control and limit these costs This is one of the reasons for using outside partners
DACOM	o o	Very high costs, especially relative to their size Long term payoff
DEC	o	Long term benefit expected; short term benefit hoped for
HP	o o	Pulling together the client/server process model cost up to $1 million in addition to uncounted staff time Benefits are immediate once the learning curve is out of the way
IBM	o o	Cost is unknown Near term and longer term benefits expected
Oraclé	o o	Cost is unknown Viewed as long term investments

11. SUCCESS/FAILURES

Cambridge Technology	o o	They see their process as responsible for their success. Do not admit to any failures
Cap Gemini	o	The believe their approach minimizes the risk of failure
DACOM	o o	They see their recent success as a company due to their approach Significant failures in past - became a learning experience
DEC	o	Would not discuss success or failure
HP	o o	The client/server model now leads to success There were failures in early implementations of the client/server model
IBM	o	Success has been mixed with failure; due to the fragmented nature of these efforts lessons or patterns are hard to draw
Oracle	o	Out of Oracle's public failure in introducing applications products came the changes that are now producing success.

12. OTHER ORGANIZATIONS TO WATCH

Cambridge Technology	o o	TI Andersen
Cap Gemini	o o	Big 6 Andersen
DACOM	o	James Martin Associates
DEC	o o o o	James Martin Associates KnowledgeWare Andersen Cambridge Technology
HP	o o o o	Dupont Farmers Insurance Microsoft Andersen
IBM	o o o o	Microsoft Next TI KnowledgeWare
Oracle	o	Several corporations (confidential)

13. CHARGING FOR HIGHER VALUE SOFTWARE ASSETS

Cambridge Technology	o	Believes it would be difficult to charge a premium
Cap Gemini	o	A premium might be possible in a long term relationship
DACOM	o	Do not think that customers will be educated enough to pay a premium for many years
DEC	o o	Not sure how premium could be computed Would be necessary to guarantee the higher value in some way
HP	o o	Doesn't believe that higher value assets will in fact be created New kinds of assets may be created
IBM	o	Not sure how additional value can be made clear to the customer
Oracle	o o	Believes asset value of software will decline because of shorter life "Information assets" may increase but unclear how to measure value

CHAPTER IV:

Summary By Category

Changes in the Software Building Process (Overall)

Cambridge Technology

They have introduced changes to bring about more rapid development and will add to or refine them in the future. They have speeded up the development process or time to develop an important or strategic application system by a factor of 3 (time to completion and use of the system) compared to similar projects at a client or other company where traditional methodologies or even some newer methods were used. (This appears to be so, but is based on using people with considerable knowledge and experience employed who impose a methodology that forces higher level user executives to provide timely and meaningful contributions to the process.)

The process allows the cost of the software building process to be reduced (including the person months involved) and the quality to be increased.

The development technique includes a well planned and executed preliminary requirements definition, rapid systems prototyping methodology to pin down requirements and parallel development teams to implement the system. (Top user executives are forced to provide timely and dedicated reviews of the results of the prototype through the activities of CT.)

Cap Gemini

They have ongoing programs concerned with improving the software building process, although their main concern is satisfying the needs of clients in specifying, delivering and installing business systems. They focus attention about improvements in the software building process on activities of software engineering firms such as Intersolv and Bachman who have been business partners.

Changes in the process which would speed up the development of business solutions, increase quality and/or reduce costs could increase competitiveness and be important to us. Improvements are being obtained on a regular basis, but they would like to see a substantial or notable change that could reduce the time for large, complex projects by 25% or more. This would gain attention from clients and prospects. However, control of change to their process is most important.

DACOM

The current approaches, especially CASE, haven't just reached a dead end but have lead us down the wrong dead end: The problem isn't producing code efficiently and correctly, but designing the right systems in the first place.

DACOM began as a CASE-oriented, essentially product, company in the early 1980s. It built reasonably good tools and was (and is) a leader in repository thinking. However, over time DACOM saw that the chief need was for an understanding of business processes and making improvements in the systems which support business processes.

DACOM has evolved from a company with an emphasis on software building to one that works closely with clients to design the right system in the first place. Software "implementation" is viewed by DACOM as almost easy once the underlying business design is set.

Overall time to complete may or may not be shorter compared to conventional methods. Much depends on how well their clients have been educated in the DACOM method. Initial projects are usually slower because of education and cultural change. Subsequent projects are probably shorter; however, this is very difficult to measure since historical measures of time to complete don't do justice to the time spent on the up-front business engineering phase.

Actual software development is much faster after DACOM's work:

- There are relatively few downstream changes
- The "specs" are unambiguous

Quality is immeasurably higher, although, again, there are few benchmarks and measuring methodologies for quality. This is very frustrating for DACOM because they and their clients strongly believe this to be true, but it is very difficult to prove. Other vendors make similar claims -- which DACOM doubts are true -- and DACOM's real achievements (to them) get lost in the general noise level.

DACOM admits that they could do a much better job at quantifying benefits, in order to

- Measure their own progress
- Assess the relative strengths of different approaches
- Market themselves better

Right now, too much of their acceptance is based on clients buying into the DACOM "religion". This is ironic and frustrating for an organization that views itself as a rigorous, engineering-oriented company.

Digital Equipment

Changes are being considered or made to speed up the development of software as well as to reduce its cost and improve quality. Many separate activities are taking place ranging from steps in consulting and systems planning to work at all levels with tools and methodologies. Specific goals have not been quantified or can't be discussed, but it seems that the speedup being sought at this time would reduce development time by one third on larger projects.

Hewlett-Packard

The driving force behind changes in the systems building process at HP is to reduce the time required to deliver projects while improving the overall quality of the finished product. HP's area of emphasis is on the management process for building systems, not on tools. They are placing emphasis on the re-engineering aspects of the business process, and simultaneously trying to develop a methodology which will apply a rigorous discipline to the resulting systems implementation.

They do believe that tools are important. However, they believe that 90% of the errors occurred at the front end, where few tools are of much use without an overall re-engineering of the business process.

They believe that their emphasis on re-engineering focusing on downsizing, organizational changes, process re-design within reasonable cost containment guidelines of the client, is the primary thrust of their rethinking of the systems development process.

Reduced cycle time on the completion of business systems re-engineering project, including the generation of finished applications systems - 25%. They would not give a specific example.

HP is trying to define quality as "fit to business need" within the client's anticipated time frame. Their position is that counting bugs in the code is secondary.

HP has not made much progress on reusability. Reusability per se is not a goal. Studies that they've conducted indicate that there is huge waste in the current process which could be significantly reduced if a set of standards could be developed along with a process to better manage code; identify and select modules to be developed and maintained as "reusable", etc. However, they feel that the process is a long way off.

The main problem is that in order to generate reusable modules, the development process must be integrated or contain components and standards which insure reusability. They think that most firms are farther behind in this area than they are in CASE. (In HP's view, most firms aren't very far along in the effective integration of CASE into the development process.)

Very little is being done to obtain any more meaningful quantification of benefits than was done in the past. This process becomes even more complex when significant business re-engineering is involved. In some instances various components could be readily quantified, but that inmost instances where business investment on the basis of much softer criteria.

IBM Corporation

There are changes being made by groups in a number of areas including professional services, external consulting, SI projects, ISSC and internal software groups. IBM's interests include business systems planning, project management, methodologies and tools. Separate activities are also being carried out with business partners.

Internal professional services and software groups are under pressure from users to speed up the development process or time to develop an application system. The reduction of development time by 50 to 100% is mentioned by users as an objective although any meaningful reduction in the overall process (e.g., 20 - 33%) would be well received. IBM recognizes these objectives, but does not have a specific target. However, IBM recognizes that changes of this magnitude will require consideration of all aspects of the process from concepts of business analysis and re-engineering to project management and code generation.

Users also want to reduce the cost of developing software for a system. The reductions being sought by users are at least 30 to 40% and as much as 100% Although there is not any official target, IBM research in Endicott, NY has talked about helping users achieve a reduction of 25 to 30%.

Another objective is to allow the time and cost required to modify software to be reduced.

A different type of objective is to develop software with a higher level of quality. In addition to programs that can be more easily modified and understood, two concepts were mentioned in regard to this. Users have stated that they are interested in systems building that would automatically incorporate various types of input data validation and other cross footing that is suitable for the type of application being developed. The second concept involves enriching the development process or adding tools which will check out the logic of software during the process and leave a structure in place for acceptance testing and testing of modifications at a later date. Auditing should also be facilitated.

Many ideas about quality are prevalent. One recommendation mentioned by users is to ensure that developed software should not be fragile or become fragile when normal types of changes are incorporated or added. Users also feel that procedures or tools should guaranty that software modules can be reused or used as an object(s).

Oracle

The principal changes taking place in the process are driven by three fundamental driving forces.

- A migration from IS to the end-user as the leading player in the development process.
- The maturation of a new life cycle based on rapid prototyping, utilizing increasingly sophisticated tool sets.
- A growing demand for rapid turnaround and responsiveness for changes to whatever system is in place.

Both downsizing from the mainframe and upsizing from standalone departmental systems to departmental work group systems is occurring simultaneously. In either case, the traditional life cycle that was used for mainframe development is or will be replaced by the cycle described below.

- Rapid prototyping of a systems basic functionality. (Joint effort of specialists - either vendors or IS - and end-user experts.) The prototype then goes into limited production.
- Refinement and rollout of the full blown system. (Joint effort of specialists and end-user experts, and based on feedback from operating the prototype in production mode.)
- Integration of the refined system with related systems. (Primarily a specialist implementation activity based on requirements identified while operating the refined system in production.)

In other words the traditional concept of a formal requirements phase is replaced by three less formal requirements definition phases. In fact each phase of the traditional life cycle has a three counterparts in the new cycle; design, development, testing, etc.

Oracle sees itself as being involved in this process in several ways:

- As an innovative tool supplier
- As a teacher/coach
- As a high level consultant and/or implementor
- As a developer of applications software (for internal use or, perhaps, in partnership with a client)
- As a supplier of flexible applications (where Oracle may again become involved as a teacher, consultant or implementor)

These are complex roles and Oracle is the first to admit that everything isn't sorted out yet.

Meeting the real business requirements of the end-user, and shortening the time from incepfion to delivery in order to capture user benefits early. This fundamentally comes down to:

- Reduced training time
- Shortened development cycle to first usable version of the system
- User self-sufficiency in terms of on-going modifications, etc.

Oracle does not believe that there is likely to be a reduction in systems development costs (at least in terms of reduced manpower) over time. The more likely scenario is that implementations will have four to five times the functional capability for the same cost.

Oracle believes that the creation of new interactive applications using their redefined life cycle and tools can be accomplished in 70% of the time it would take to achieve the same functionality using conventional CASE and methods. For batch based applications or the generation of new reports, etc., they believe the number is 25%. (See attached diagram showing how newer approaches to system building generate benefits.)

Quality is customer satisfaction with the provided functionality, and "0 defect code".

In a homogeneous environment (Oracle Platform) they estimate that a fair amount of the code can be re-used. But this is not an objective unto itself. In heterogeneous environments (say using DB2 along with Oracle), less will be achieved in terms of reusability.

Oracle believes that the industry is a long way off from obtaining reusability on any major scale through the development of "objects". Probably the highest level of reusability comes through the creation of user "customizable" applications software products.

Very little quantification of benefits has been done. Most end-user projects (which are an increasing proportion of the total) seldom undergo a formal justification or subsequent audit process.

Soft Benefits

Cambridge Technology

The use of rapid prototyping during the requirements definition helps to improve communication about requirements as well as to analyze what is going on in business processes. If business systems are being re-engineered, concepts can be tested out. The prototype software enables users to review systems and gives them a feeling of satisfaction with the final implementation.

Cap Gemini

Improvements in software building should include steps to aid users to review how business processes will be changed or re-engineered as well as to review data management and utilization. Improvements in software engineering tools have aided both of these objectives.

DACOM

DACOM would take issue with the concept of "softer" benefits. The benefits are softer only from the standpoint that many (maybe most) of the benefits do not accrue to software building, but are gained by the underlying business. (DACOM agrees that it would be very powerful to measure the "before" and "after" business costs and benefits, but it does not do so on a consistent basis. Its clients don't want to pay for this kind of activity, because doing so won't help solve their particular business problem.}

DACOM works largely with true end users in the manufacturing/logistics area. From the standpoint of DACOM's clients the current system building process is "intolerable" and not just for the usual reasons of lateness and expense. The more thoughtful clients (and DACOM appeals to thoughtful clients, for better or worse) are frustrated by software that

- Doesn't meet their business functional needs very well.
- Behaves in inconsistent and unpredictable ways; users are overwhelmed by system details; computer systems make it harder rather than easier to understand underlying problems.
- Is difficult to change in the terms of time and expense and, even then, often can't really be changed to meet user needs

Downsizing of organizations and of the computing environment multiplies these problems

Their clients don't really care about the usual measures used to measure a "good" software project.

Digital Equipment

There is a definite need to achieve other, softer benefits for customers including improved satisfaction with the development process, improved communications on the work being done, and more ability to respond to needs. These needs are being investigated in relation to systems integration and professional services work as well as in relation to internal analyses of systems building.

Hewlett-Packard

HP's experience has been that the two dominant soft benefits that firms are using to justify re-systemization (and business re-engineering) efforts are:

- Improved customer satisfaction; and
- Reduced cycle time on core business transactions

Certain of HP's customers have taken to trying to measure both of these factors; but it's still difficult to relate the measures to the bottom line performance of the firm.

IBM Corporation

Software should facilitate efforts to communicate as well as to analyze what is going on in business processes. There should be a view(s) of software that enable users to review systems and give them a feeling of satisfaction with the software.

Oracle

As firms begin to leverage the concept of information distribution and collection utilizing the interactive capabilities of modern technology significant benefits will be realized in terms of better communications both deferred and direct. This will include interaction between individuals in a firm as well as between an individual and the corporate repository of knowledge. Applying these same systems design concepts to interactions with customers, etc. will bring another whole set of benefits.

The typical knowledge worker spends between 10-15% of his/her time today interacting with paper. As the level of integration of systems goes up and the ability to interact on an as needed basis with the corporate repository grows, knowledge workers will spend probably 20-25% of their time working with information electronically.

In Oracle's view we're just seeing the tip of the iceberg in terms of understanding, articulating and realizing some of the significant benefits that can be achieved through the migration to interactive systems at all levels.

Relative Contribution of Tools and Processes

Cambridge Technology

Both tools and processes are used in the CT Partners methodology. The rapid prototyping capability is more of a tool-like technique that allows concepts to be tested.

The development of the implementation code by the use of parallel teams is more of a process or factory like method in which one team handles the application, a second handles connectivity, a third the user interface, and a fourth the data.

Cap Gemini

As noted above, tools have brought ongoing improvements in the software building process, but changes in the overall process have made the greatest impact on their ability to deliver timely and quality work, and we would like to see an improvement in the process that would provide a substantial improvement in time and cost requirements while also enhancing quality further.

They are reviewing the use of factory-like methods to generate code although most of the effort is devoted to working with users and difficulties could occur in trying to implement a factory process in that situation.

DACOM

DACOM has a full perspective, having started as a CASE company (before the term was invented) and now describe themselves as a "systems integration consultant."

DACOM does use tools, some of which are their own (see attachment 1).

However, they see tools as essentially supporting a model-oriented approach ("Tools are the finger painting of business engineering").

- They see almost all CASE tools as not supporting the improvement of business processes (although pieces of them can do so with the right orientation)
- A good repository is vital, but they don't know of one that really meets their needs.
- They are forced to use and link standalone tools. They hope that in the future there will be integrated tools that will support a range of modeling techniques.

A key component (and bottleneck) is the need for consultants with the necessary "wisdom" to integrate the business problem, modeling techniques and client change. This is the strength of their business, but also a key limitation.

- These traits make them valuable to customers, but expensive.
- DACOM would like to be able to leverage this knowledge in a more embedded methodology. They see the application of AI and fuzzy logic as possibilities, but are currently resource-constrained in being able to pursue these directions very aggressively. (They are especially cautious since in the past they have tried to prematurely productize processes.)

The general process that they use is shown in Attachment 2.

Digital Equipment

Tools including CASE tools and other software aids are being sought and evaluated to aid with the system building process. Tools may seem to be more important in regard to current improvements in systems building, but improvement of the entire process that is carried out in systems building offers more opportunities for improvement.

The use of techniques such as JAD or RAD as part of a process are expected to have more impact on time, cost and quality than tools can.

Research is being conducted that addresses both the use of tools and processes to improve the system building process. The use of manufacturing techniques in software production has been carried out and is also being investigated together with other topics.

Hewlett-Packard

HP believes that the biggest contributions are coming through improvements in the process and people management aspects of the systems building process. CASE tools and other automated approaches are important but secondary. They have adopted an approach which pairs top-flight project managers with tailored teams for specific implementations. Project managers are responsible for projects only. Team members are managed in the personnel and development sense by other managers within their home organizations.

- They have installed new processes that permit feedback between project managers and those managers in the firm who are responsible for the development and administration of individuals within the organization.
- They have installed a special program to continually improve even seasoned project managers skills.
- They are using some incentive programs for project performance and quality on an experimental basis, but these programs are immature in that they tend to reward heroic fire fighting efforts...rather than solid fire prevention on a day to day basis. Incentives (both team and individual will probably play a stronger role in the future.

IBM Corporation

CASE won't achieve all the objectives that are being discussed for software but CASE tools can aid by enabling software to be analyzed, modified and generated more easily for a number of applications. It was noted that certain applications that require high levels of transaction processing might require attention from personnel that were experts in the alternatives and tools available to meet tasks. One project mentioned in this regard involved an RS/6000 system that AIC worked on in conjunction with the research staff in Austin.

An expert on GUI has been hired by IBM in Endicott, NY to develop special tools to support and guide developers in implementing certain types of workstation systems faster and with more quality.

In regard to the use of tools, it is felt that AI and particularly expert systems, as well as other capabilities should be able to be incorporated as objects in the software development process.

It is felt that the process of systems development must address a much broader set of needs than tools can address. For instance, it must address changing work structures such as work groups and other changes in work organization or people management.

Ideas for the application of manufacturing techniques to software building are also being considered, tracked and experimented with although it is felt that it may be difficult to achieve other objectives such as improvements in communication as mentioned above together with software manufacturing.

Oracle

New tools and changes in processes are probably making an equal contribution; and they continue to evolve in an iterative manner. The methodology that Oracle endorses of rapid prototyping, refinement and integration has stimulated the need for new functionality in the tool set. And, the capabilities that are subsequently generated in the tool set provide opportunities for further refinement in the design and implementation processes.

CASE tools are either obsolete or essential. If they are integrated with rapid prototyping methódology and deal with the concept of distributed architecture, they are essential. Tools that "assume" the kinds of structured methodologies of the 70's and 80's are probably obsolete because they don't integrate with current methodologies or architectures.

AI, like many other buzz words (including CASE), has gotten a bad name. However, if you look around, you'll see that expert rule based systems are imbedded in everything from spreadsheets and word processors to systems development tools. The use of these technologies to support the systems building process will continue to grow, and will result in both reduced development time and higher quality code.

The biggest single change on the people and organization side was the migration of development responsibility from IS to the end-user. Distributed IS has changed the process. The result is more teamed approaches which hopefully combine the required technological expertise with user knowledge of business process.

In terms of management techniques, people are still going to be people and managers will still vary in quality. However, the teaming of users and technical personnel in the development process will and should continue to reduce the complexities of management by improving communications. New tools and processes will continue to reduce the number of people required to accomplish an implementation (and therefore the complexity).

Oracle believes there is some fallacy in the notion of "manufacturing" code. Manufacturing implies making a large quantity of the same (or essentially the same) thing. Building applications involves using common processes to generate unique systems. The thing that they have in common is the use of consistent (but obviously different) processes... little else.

Reusability

Cambridge Technology

The methodology described above has not focused on reusability, but the use of parallel development efforts would facilitate the consideration of reusable code since functional aspects of coding are separated.

Cap Gemini

Cap Gemini and its business partners are exploring the use of reusability of objects as well as object oriented programming, but this is one of many efforts to improve methods.

DACOM

They are attracted to the concept of reusability but have found only a small number of "paradigm processes" which they can see as potential candidates. For example, "inventory" should qualify, but doesn't once the major differences between industries are examined as well as the "flavors" between firms.

They see no theory of reusability in the commercial world (unlike the formulas in the scientific world) to make their job easier. They use common skeletons and analyze differences but don't really view this as reusability.

They reuse "wisdom" and are aware of the deficiencies in doing so.

Digital Equipment

Tools are being used that allow a repeatable solution to be sold. For certain applications, these tools will allow the solution to be modified or customized more easily, and many components of the solution would not have to be changed. There have been various types of reusability in the past by DEC VARs as well as by DEC. Digital has also been experimenting with reusing objects in a group of internal projects and is now doing that on a regular basis

Hewlett-Packard

Very little is being done by HP in the reusability concept as applied to custom code development. However, a significant amount of work is going into refining process which will insure reusability of some significant portion of the code that's developed for applications or systems software products.

What is being done in this area is proprietary to HP and its products. They are still a long way from the point where the generation of "universal objects", etc. will be commonplace

IBM Corporation

Reusability was mentioned several times as an objective in relation to the goals stated above. The objective was stated as creating software that can be reused and/or facilitate the reuse of other software whenever feasible.

Ideas have been tried out internally by IBM in Boca Raton, Austin, and other locations to reuse software during systems software development.

Research projects have also been carried out to test reusability and other concepts and to track user and vendor work with reusability.

Oracle

From Oracle's point of view, reusability is an objective. However, a lot of effort will need to be placed into the development of standards, and the consistent use of sophisticated tools before this will be accomplished through "object" libraries. In the meantime, Oracle targets on being able to reuse higher level application definitions to "regenerate" code. Modifications to the definitions at the high end allow customization; and the use of tools allows new custom code to be generated at minimal cost.

There may not be much progress was likely to get made in generating reusable modules at the user interface, since this is an area where systems providing similar functionality were liable to vary significantly based on personal and organizational preferences.

Difficulties in Addressing Client/Server

Cambridge Technology

There can be difficulties since corporate objectives as well as processes and data can be divided between uncoordinated systems on client/servers. Cambridge Technology allows client/server systems to be handled by themselves or better yet, together with other, related systems changes that are necessary to meet corporate goals. The methodology utilizes open systems which facilitates work with a variety of client/server technology.

They also test out application approaches with rapid prototyping that should highlight questions about the segmentation of functions and data across systems.

Cap Gemini

The needs of users have to be fully understood before decisions are made regarding the technology that will be used in a solution. Decisions are made too quickly in regard to the use of client/server technology, and problems can result in relation to data management and the division of functions between business units.

DACOM

The client/server model is extremely incomplete. More basically, there is no model for organizational or functional downsizing. This puts a process engineering in a double bind. Learning for DACOM is "painful" and largely intuitive as far as where data, processing and location boundaries are drawn.

This disturbs them less than may appear, since the analytic process involved in understanding the business processes will usually keep them from making unrecoverable mistakes in "client" and "server" function allocations.

They feel great sympathy with their partners (whether customer-partners or vendor-partners) who are trying to make client/server technology work.

Digital Equipment

There are questions about the allocation of data and processes from a consolidated system to a number of client/server systems. There are also questions about using a set of data or transactions received at one point to update multiple client/server systems.

Hewlett-Packard

HP has been working at the C/S model in an open systems environment for some number of years. They are very comfortable with it at this point, and have a formally documented approach for allocating functions and data between clients and servers. (As is the case with reusability, the model is proprietary to HP and might not be applicable to other platform architectures.) The model is supposed to be rigorous and has significantly improved the quality and efficiency with which new applications can be developing using C/S.

They still needed to improve the overall process at the design level. The area that needs the greatest improvement is a consistent approach to linking redesigned business processes with functional specifications for C/S systems. They are working on this.

- They have just dropped KnowledgeWare in favor of IEF as their primary CASE vendor.
- Anticipate that IEF organization will work jointly with them on tailoring CASE tools to meet HP's needs both at the design and implementation levels.

A formal model for the C/S design process has significantly improved the quality and reduced the time required to systematically test new code. Estimates that they are saving about 20-25% in this area, and are certain that the quality of the code is higher.

IBM Corporation

Difficulties involved in separating processes or splitting the management of data have not been fully addressed. IBM thinks that these problems have often led to the use of larger capabilities, AS/400s rather than work stations, in many client/server systems.

Oracle

Oracle believes that the C/S model is incomplete. When utilizing a consistent platform on any given set of applications (the Oracle platform) they have a consistent way to model the distribution of processing and data. It's pretty straightforward. Shared functions and data go to the server, and unique functions and data go to the client. However, when working with heterogeneous platforms (in particular data bases), the ability to follow the model is frequently inhibited by differences in functionality in hardware and software.

Over time, the migration to more open systems and standard interfaces will likely increase the level of consistency that can be achieved in allocating data and processing in the C/S environment. In any event, its not a serious enough problem that it is likely to inhibit the ongoing emergence of C/S as the primary applications platform.

Is More Than One Process Model to Improve Systems Building?

Cambridge Technology

CT Partners uses only one process model, although improvements to it are considered, evaluated and implemented over time. They have a high degree of control over management of change.

Cap Gemini

Within CGA, there is an effort to use one model although experimentation is made with other models. We are willing to support projects at users that employ models different from the one we support internally. Some individual offices or personnel are attempting to change methods, but control is achieved. They have a medium-high degree of control over the management of change.

Due to uncertainties about the systems building process as well as the lack of coordination between various department in a company, multiple models may be in use at a client, and they might be involved in supporting more than one.

DACOM

They have their own approach (attachment 2). However, they feel agnostic toward particular business analysis practices and feel it is too early to back one horse. They have a high degree of control over the management of change.

Digital Equipment

DEC is using multiple process models, but most of these efforts are coordinated. In some circumstances, activities in support of accounts could use tools and techniques or new methodologies in an unplanned manner to improve systems building. Where a VAR or associated vendor is in a lead position, this is particularly true. They have a medium degree o control over the management of change.

Tools are available to help manage or control change management at DEC, but they are not always used.

Hewlett-Packard

HP is definitely trying multiple approaches to improving the systems building process, but the process is not seen internally as very well coordinated. Individual initiatives appear to pop up, and get piloted. Informal communications between people involved in these efforts are the primary method of coordination. However, enough information gets shared to insure that some "best of breed" approaches are emerging.

There is a group manager whose responsibility it is to gather, digest and redistribute information on the new techniques and approaches that emerge. Overall, they have a medium to low degree of control over the management of change.

IBM Corporation

IBM has a number of process models available to improve systems building and is constantly in contact with other vendors to make arrangements to share and try out their ideas (e.g., recent agreement with HP on client/server development products). They have a low degree of control over the management of change. Many offices (but not all) are supportive of change due to this. IBM contacts in professional services assignments feel that this emphasizes that the most important element in the improvement of systems building is the experience and knowledge of industries and application systems.

IBM points out that some large corporations such as several of the top banks are trying to use more than one process model for improving systems building. This has occurred chiefly due to unplanned pressures by users for different alternatives for systems building. There may not be an adequate means of managing or coordinating change in these situations even if IS tries to coordinate or manage change in situations where more than one model is operative.

Some areas of organizations are more supportive of change, particularly users who are under pressure to achieve business goals.

Oracle

Yes, Oracle is certainly trying more than one model. Two examples have already been cited; the rapid prototyping process and the creation of "customizable" applications packages. More are likely to be tried in the future.

In terms of how the process is being managed... at Oracle it's ad hoc. Overall, they have a low degree of control over the management of change.

Skill Needs and Distribution

Cambridge Technology

Higher level skills are in short supply and distributed unevenly. CT Partners provides a method that allows the need for skills to be managed and controlled more easily.

The CT Group, an affiliated organization, provides education in the methodology utilized by CT Partners which provides more opportunity for overcoming skill deficiencies.

Cap Gemini

There are currently needs for more software development skills due to the variety of tools, languages, databases, operating systems, and other systems and application software in use for workstations as well as other platforms. CGA as well as other SI and professional services firms can not count on having the skills available that prospects and clients have in use or plan to utilize. Alliances, temporary staff members and the services of specialized temporary employment services must be used.

Skills bottlenecks are being encountered. CGA will propose changes in the software products planned or specified for jobs by users. Recently, one client was persuaded to step back to an earlier version of a workstation operating system in order to use the network manager which had been selected.

In addition to proposing changes to the set of software products selected for a project, the proactive step more often taken by CGA is to work with clients and recommend a set of products that can be well supported.

Internal education as well as relations with consultants and other firms are also sought to expand the skills inventory that CGA can support.

DACOM

Finding people who understand business processes is their greatest need. It takes several years to train a new employee with little business experience (largely on the job). They can predict fairly well the people who will not ultimately make it, after a day of pre-qualification testing; however, they can only predict those who will not fail; they are much less sure of those who will do very well. They have often been forced to hire more senior people as a result; this kind of person is more expensive, pound for pound, than a more junior person. A larger problem is that it is more difficult to form a cohesive "DACOM way" from people who are more set in their ways (even if these are sound, productive ways).

DACOM would like to emulate the philosophy of an EDS or Andersen and mold new recruits to their culture.

Digital Equipment

There are a number of new skills that are needed in current projects ranging from those needed to use new systems development and relational database tools to those available to use network software products, GUI and workstation spreadsheet and database products.

The new skills are not always present at DEC. Some internal training is carried on and external training is made use of. However, a group of contract services vendors are used to meet critical situations with temporary people. VARs and associated consultants are also used to meet critical needs.

Hewlett-Packard

Technical skills don't seem to be a problem for HP. The company's culture supports investing in employees, particularly in technical education. Project management is the primary area where they see a significant shortage. And, there are no magic wands to wave that eliminate the need for the "experience" factor in project management. They are trying an "understudy" approach which pairs less experienced project managers with more experienced ones to hopefully obtain knowledge transfer.

IBM Corporation

Skills are still in short supply and distributed unevenly. Hiring restrictions and staff reductions have made it difficult to get the right people. The IBM culture still places great emphasis on training. Proactive steps to improve the situation include arrangements with vendors that can aid with or participate in improvement in software building.

Oracle

Oracle, being a technology company, makes heavy investments in training, and provides numerous incentives for self-development. On-going training and development is just part of the culture. So fundamentally, Oracle doesn't feel that it's facing a any significant problém in finding or developing the skills that it needs.

In general, Oracle sees the distribution of IS skills to user organizations is a significant change, and will continue to happen throughout the decade. In terms of skills bottlenecks, Oracle believes that these are largely transient effects.

System Testing Process

Cambridge Technology

Users are being brought in to systems testing more, particularly to review the systems concept or business re-engineering envisioned. The methodology of CT Partners is supportive of this by enabling users to formulate requirements in a structured manner and test out the system approach more quickly through the rapid generation of a prototype.

Cap Gemini

More comprehensive tests are formulated with the aid of users, and users are brought into the review of proposed systems at an earlier stage through the use of front end CASE tools and JAD or RAD types of processes.

DACOM

Testing should be done at the model level.

Digital Equipment

Efforts are being made at Digital to develop new types of testing that will be automatically incorporated into the system building process.

There are also some products being introduced jointly with large companies that will handle portions of the testing process in new ways through system review with front end CASE tools, prototyping and other means.

Hewlett-Packard

Adoption of the C/S model has significantly changed (and improved) the testing process. (See the answer to question 5.) Since the modes provides for a limited set of structured communications between client and server using standard protocols, it's possible to develop test packages to automate the testing of each units functionality separately and in most instances simultaneously. This has lowered testing costs and cycle times significantly.

IBM Corporation

Users are being brought in to systems testing to a greater extent. New types of tools or systems review are being sought to improve the ability of users to test systems. Where vendors are involved in the supply of or development of software products, more accountability is being placed upon them during systems testing.

Oracle

Testing is becoming an on-going continuous process. The objective is no longer just to meet a written specification, but to insure that the customer is satisfied with the cost, functional performance and time to delivery of the system. In a sense this increases the complexity of the entire testing process, and places the "pass fail" judgment directly to the user.

In terms of actually testing finished code, major tests are built in at each of the three major phases of the Oracle process (rapid prototyping, refinement, integration).

The process of regression testing of software is growing more complex. As more and more functionality is included, the possibilities that need to be tested multiply exponentially. Some automated processes have been put in place, but manual testing of individual products is still required.

Particular Innovations

Cambridge Technology

CT Partner's believes its methodology (described in section 1) is particularly innovative.

CT Partner's also feels that its ability to rapidly develop strategic systems using an open system approach is innovative.

Cap Gemini

The effort to use the software engineering tools and concepts of leading software engineering firms together with the experience of CGA is felt to be innovative. Some other SI/professional services firms concentrate on using their own tools and methodologies. CGA wants to incorporate the ideas of firms who specialize in improving the software building process.

Position papers and internal training ensures that innovations are spread throughout CGA. CGA is prepared to work with clients to help them speed innovative changes in systems building, but this can be difficult to achieve in large organizations, since many user groups are acting independently.

DACOM

DACOM believes that their entire approach is innovative, i.e.,

- Emphasizing modeling the upstream business process and defining objectives
- Focus on model building
- Making customer part of team, involving high levels of education and training
- De-emphasize conventional "software building" part of the equation (even though many of the DACOM staff has a heavy background in software building).

Digital Equipment

Digital feels that its system building process is innovative since it results in complex network solutions in a relatively short time.

Several recent internal efforts have resulted in factory-like development of code, but can't be discussed yet.

Hewlett-Packard

They believe that there is significant variation between major firms, but that probably all leading firms have hot spots where advances are being made. Several user companies and two vendors were cited.

- Dupont - Significant work in "rapid" prototyping for C/S applications
- Farmers' Insurance - Advanced architecture for C/S transaction management
- Microsoft - Use of advanced tools and concepts (object oriented coding, etc.) to automate the systems building process. Reusability at applications product level.
- HP - C/S applications model and transaction management
- Andersen - Re-engineering process and (probably) reusable applications modules.

IBM Corporation

Innovation of various types is being tried by corporations in contact with IBM including the development of specialized tools internally or by vendors, the generation of objects in C++ code that can be used to add necessary functions to a number of systems, the use of specialized CASE approaches, greater reliance on vendor application software products and experimentation with new types of languages. One effort that IBM is engage in with a user involves the use of specialized GUI that aid a developer to combine objects. Use of the GUI will guide (and force) the developer to include various types of controls and aids that will facilitate testing and operation of the resulting system.

IBM is also conducting research on ideas about GUI that use symbols other than icons. One of these concerns the use of symbols that would aid a person involved in a work group structure.

Oracle

- More and more integration of applications systems and data
- Installing methodologies which permit the continuous refinement of systems
- Providing tools and methodologies which allow users to achieve high levels of self-sufficiency in terms of managing their own systems environment.

Cost of Making Improvements in Systems Building

(Funds, People, and Opportunities Foregone)

Cambridge Technology

Cambridge Technology has experience in relation to this topic. It originally sold education that concentrated on using technology to obtain competitive advantage. The cost of implementing the methods it advocated were hard to justify in many organizations. It was easier for these organizations to pay for the use of the methods to implement strategic applications more rapidly. A near term payoff on a project justified using the methods.

Since the current process is so controlled, the cost of making improvements to it can be evaluated in terms of both near and longer term benefits. No changes are foreseen that would change the process to a significant extent.

Cap Gemini

Improvements in systems building within CGA takes time and consume funds and personnel, and can delay getting work done and limit or preclude taking advantage of business opportunities. That is why CGA tries to use the experience of business partners active in systems development processes and tools as well as experiment with methods that might improve systems development. CGA wants to be able to work with clients that want to use various improvements, even if temporary consultants are employed to aid with that; however, CGA wants to limit internal changes unless there are longer term advantages. .

DACOM

High costs, both to themselves and their customers. [A strength and weakness of DACOM is the fuzzy line between themselves and their customers.] DACOM has plowed back a great deal of time and money into iterations of their approach. They believe that they are on a very productive path now. But the whole process has definitely been a long term investment for them.

Digital Equipment

It is hoped that there will be some rapid or near term improvements although most observers at DEC feel that it will be a long term investment.

Hewlett-Packard

The following is HP's educated guess:

- In addition to whatever investment cost their might be in designing a new process or technique, the first time the process is put into use on an actual systems effort, there is probably a twenty percent penalty in development cost.
- On the second attempt the penalty probably drops to 10% and on future efforts gains in productivity can be made in the order of 10-20% per effort.

To pull together HP's C/S model probably cost between $500K and $1M. This doesn't count the many hours that were invested in developing stand alone components which were used in the model. However, benefits are there from the developers viewpoint.

After two years of working with the model in actual development efforts, they are probably delivering systems at 80% of the cost for these same systems developed using conventional methods.

IBM Corporation

The cost of making improvements can not be tracked on an overall basis in IBM and is not being tracked in most organizations. IBM is willing to make improvements to obtain near term business or to improve capabilities over a longer term. It is felt that many short term efforts should be undertaken in order to ensure that IBM will be aware of the more outstanding improvements.

Oracle

Unfortunately this is something that isn't measured. Partly because what is being done is intertwined with so many other activities, it's not clear that even a heavy investment in the measurement process would produce a result of significant value. However, major investments in new methodologies and tools were looked upon by users as long term investments. Only when the first applications came rolling out under the new approach would people begin to realize the potential savings or pay back.

Success/Failures

Cambridge Technology

The use of the rapid prototyping technique to review system approaches and how systems will work has helped CT Partner's avoid real failures. Past experience has taught the organization that its methods must be adhered to in order to guaranty benefits.

Cap Gemini

CGA tries to avoid failures in using new systems building methods by working with business partners who are developing and testing out the methods.

DACOM

There were several major failures in business concept and execution in the mid-1980s.

- They were overly product oriented
- They did not sufficiently value or leverage their "wisdom" in the past from a business standpoint

Digital Equipment

There have been difficulties encountered, but failures have not been recorded or won't be discussed.

Hewlett-Packard

The C/S development model has been Hp's biggest success. On the failure side, early on they had an occasional disaster in terms of actually using the model to do a client implementation.

IBM Corporation

Failures or the inability to achieve total objectives has been notable in some CASE projects as well as projects run by IBM that were supposed to improve software building. A major lesson that was learned was that the business objectives and processes are sometimes neglected or not reviewed sufficiently and rethought during projects that automate systems building.

Oracle

It's no secret that Oracle's biggest failure was in its initial approach to the applications software market. The products (originally developed for internal use) were low in quality and functionality, support was low quality, and sales commitments exceeded the deliverable capability. This problem is behind them now, and caused a major rethinking about the entire applications building process which forced Oracle into its new direction - with benefits all around.

Other Organizations to Watch

Cambridge Technology

A number of systems integrators are improving their ability to build systems, but no one has taken the steps that CT Partners has to both speed up development and review a model of the business system before it is delivered.

Texas Instruments as well as Andersen are known for efforts to improve systems building.

Cap Gemini

Leading systems integrators such as big 6 firms and CSC are always trying to improve their ability to build systems. Andersen Consulting stands out as a firm that they believe has achieved improvements.

DACOM

James Martin Associates are doing good work. Otherwise, they admit to being inward-looking and do not "follow the competition". To some extent this is because they do not see most other vendors as being in the same kind of business as they are. That is, vendors that are working on software development improvements are not viewed as having many lessons for them. They view most other "business process re-eningeering" vendors as being much more touchy-feely (or management consulting oriented). But they admit that they may be wrong.

Digital Equipment

These range from organizations concerned with improvements such as James Martin & Associates and KnowledgeWare to vendors such as Andersen Consulting and Cambridge Technology. Andersen has a methodology and tools that have been demonstrated on a number of projects.

Hewlett-Packard

HP is probably ahead of the average firm in rethinking and actually implementing changes to the software building process. They believe they were probably far from the head of the pack. They don't think that there was any one firm who was the leader in all aspects of the process.

IBM Corporation

The vendor mentioned most often as one who is or may be achieving such improvements is Microsoft. They have been building and rebuilding software objects so that they can be joined together more easily. Visible Basic has been improved and re-coded in C++ to provide users the means of reviewing and adding to or modifying objects in a more easy fashion. The Next operating system of Jobs was felt to be a good step in the direction of improved development, but is not spoken of lately. Products such as Powerbuilder and the Insynch Passport system that generates C code are also being reviewed in addition to the products of TI, KnowledgeWare, Bachman and others whose products are under constant review.

Oracle

Some significant improvements are being made in the pharmaceutical industry. (Examples which confidential.) There are similar developments in the chemical where the adoption of new techniques and approaches to systems is allowing high levels of integration improving overall productivity to the firm.

Charging for Higher Value Software Assets

Cambridge Technology

A systems integrator can obtain recognition and sufficient remuneration, particularly when steps such as the use of a rapid prototype are used. It might be hard to translate this process into building software products that commanded a premium price over other software products however since it could be hard to convince buyers that products offered a competitive advantage that would command a much higher price. Competitors would claim that they had something almost as good at a much lower price.

Cap Gemini

The total value of the assets can only be measured over time, although there may be more immediate benefits such as more rapid and less cost development or easier testing.

A systems integrator such as CGA can obtain recognition and remuneration based on the ability of delivering a solution. It might be necessary to establish a long term relationship to gain additional payment for delivering higher quality assets that would facilitate systems modification and growth for instance.

DACOM

They do not believe that their "consumers" are sufficiently educated to understand the issues involved. This goes back to the hard data question: If vendors (including themselves) could better demonstrate that this was true, then perhaps customers would pay more. They see this involving a long process of education and culture change -- 5 or 10 years at least -- before (a) This becomes true and (b) customers will accept it as being true.

Digital Equipment

Not sure how it can be measured and recognized, except over a period of time.

It will be necessary to guaranty the higher value in some way, if users are asked to pay more.

Hewlett-Packard

HP doesn't feel that the asset value of individual custom designed systems will improve significantly as the direct result of improvements to the process; but does believe that the process will create new kinds of assets.

The business process models developed as a result of re-engineering can and (in a few instances) are being sold or brokered. Repositories of basic business processes will one day be sold. Using these models through a medium such as IEF, other firms could tailor both the processes and the code to meet their particular business needs. One example cited was an airline used TI to broker its IEF model for frequent flyer operations to another airline serving a non-competitive market.

IBM Corporation

Although it is believed that higher value software assets will be developed, there are unanswered questions about the ability of a vendor to achieve sufficient recognition for the software assets put in place so that a premium can be charged on a near term basis.

Oracle

The actual asset value of software will probably go down as it becomes easier to build and is replaced more quickly. However, the value of the "information asset" will likely rise significantly as newer applications and software systems permit its distribution, collection and utilization in ways that weren't previously possible. Whether anyone can measure these values is debatable.

CHAPTER V:

Summary By Vendor

Cambridge Technology Partners

1. Changes in the Software Building Process (Overall)

They have introduced changes to bring about more rapid development and will add to or refine them in the future. They have speeded up the development process or time to develop an important or strategic application system by a factor of 3 (time to completion and use of the system) compared to similar projects at a client or other company where traditional methodologies or even some newer methods were used. (This appears to be so, but is based on using people with considerable knowledge and experience employed who impose a methodology that forces higher level user executives to provide timely and meaningful contributions to the process.)

The process allows the cost of the software building process to be reduced (including the person months involved) and the quality to be increased.

The development technique includes a well planned and executed preliminary requirements definition, rapid systems prototyping methodology to pin down requirements and parallel development teams to implement the system. (Top user executives are forced to provide timely and dedicated reviews of the results of the prototype through the activities of CT.)

2. Soft Benefits

The use of rapid prototyping during the requirements definition helps to improve communication about requirements as well as to analyze what is going on in business processes. If business systems are being re-engineered, concepts can be tested out. The prototype software enables users to review systems and gives them a feeling of satisfaction with the final implementation.

3. Relative Contribution of Tools and Processes

Both tools and processes are used in the CT Partners methodology. The rapid prototyping capability is more of a tool-like technique that allows concepts to be tested.

The development of the implementation code by the use of parallel teams is more of a process or factory like method in which one team handles the application, a second handles connectivity, a third the user interface, and a fourth the data.

4. Reusability

The methodology described above has not focused on reusability, but the use of parallel development efforts would facilitate the consideration of reusable code since functional aspects of coding are separated.

5. Difficulties in Addressing Client/Server

There can be difficulties since corporate objectives as well as processes and data can be divided between uncoordinated systems on client/servers. Cambridge Technology allows client/server systems to be handled by themselves or better yet, together with other, related systems changes that are necessary to meet corporate goals. The methodology utilizes open systems which facilitates work with a variety of client/server technology.

They also test out application approaches with rapid prototyping that should highlight questions about the segmentation of functions and data across systems.

6. Is More Than One Process Model to Improve Systems Building?

CT Partners uses only one process model, although improvements to it are considered, evaluated and implemented over time. They have a high degree of control over management of change.

7. Skill Needs and Distribution

Higher level skills are in short supply and distributed unevenly. CT Partners provides a method that allows the need for skills to be managed and controlled more easily.

The CT Group, an affiliated organization, provides education in the methodology utilized by CT Partners which provides more opportunity for overcoming skill deficiencies.

8. System Testing Process

Users are being brought in to systems testing more, particularly to review the systems concept or business re-engineering envisioned. The methodology of CT Partners is supportive of this by enabling users to formulate requirements in a structured manner and test out the system approach more quickly through the rapid generation of a prototype.

9. Particular Innovations

CT Partner's believes its methodology (described in section 1) is particularly innovative.

CT Partner's also feels that its ability to rapidly develop strategic systems using an open system approach is innovative.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

Cambridge Technology has experience in relation to this topic. It originally sold education that concentrated on using technology to obtain competitive advantage. The cost of implementing the methods it advocated were hard to justify in many organizations. It was easier for these organizations to pay for the use of the methods to implement strategic applications more rapidly. A near term payoff on a project justified using the methods.

Since the current process is so controlled, the cost of making improvements to it can be evaluated in terms of both near and longer term benefits. No changes are foreseen that would change the process to a significant extent.

11. Success/Failures

The use of the rapid prototyping technique to review system approaches and how systems will work has helped CT Partner's avoid real failures. Past experience has taught the organization that its methods must be adhered to in order to guaranty benefits.

12. Other Organizations to Watch

A number of systems integrators are improving their ability to build systems, but no one has taken the steps that CT Partners has to both speed up development and review a model of the business system before it is delivered.

Texas Instruments as well as Andersen are known for efforts to improve systems building.

13. Charging for Higher Value Software Assets

A systems integrator can obtain recognition and sufficient remuneration, particularly when steps such as the use of a rapid prototype are used. It might be hard to translate this process into building software products that commanded a premium price over other software products however since it could be hard to convince buyers that products offered a competitive advantage that would command a much higher price. Competitors would claim that they had something almost as good at a much lower price.

CAP Gemini

1. Changes in the Software Building Process (Overall)

They have ongoing programs concerned with improving the software building process, although their main concern is satisfying the needs of clients in specifying, delivering and installing business systems. They focus attention about improvements in the software building process on activities of software engineering firms such as Intersolv and Bachman who have been business partners.

Changes in the process which would speed up the development of business solutions, increase quality and/or reduce costs could increase competitiveness and be important to us. Improvements are being obtained on a regular basis, but they would like to see a substantial or notable change that could reduce the time for large, complex projects by 25% or more. This would gain attention from clients and prospects. However, control of change to their process is most important.

2. Soft Benefits

Improvements in software building should include steps to aid users to review how business processes will be changed or re-engineered as well as to review data management and utilization. Improvements in software engineering tools have aided both of these objectives.

3. Relative Contribution of Tools and Processes

As noted above, tools have brought ongoing improvements in the software building process, but changes in the overall process have made the greatest impact on their ability to deliver timely and quality work, and we would like to see an improvement in the process that would provide a substantial improvement in time and cost requirements while also enhancing quality further.

They are reviewing the use of factory-like methods to generate code although most of the effort is devoted to working with users and difficulties could occur in trying to implement a factory process in that situation.

4. Reusability

Cap Gemini and its business partners are exploring the use of reusability of objects as well as object oriented programming, but this is one of many efforts to improve methods.

5. Difficulties in Addressing Client/Server

The needs of users have to be fully understood before decisions are made regarding the technology that will be used in a solution. Decisions are made too quickly in regard to the use of client/server technology, and problems can result in relation to data management and the division of functions between business units.

6. Is More Than One Process Model to Improve Systems Building?

Within CGA, there is an effort to use one model although experimentation is made with other models. We are willing to support projects at users that employ models different from the one we support internally. Some individual offices or personnel are attempting to change methods, but control is achieved. They have a medium-high degree of control over the management of change.

Due to uncertainties about the systems building process as well as the lack of coordination between various department in a company, multiple models may be in use at a client, and they might be involved in supporting more than one.

7. Skill Needs and Distribution

There are currently needs for more software development skills due to the variety of tools, languages, databases, operating systems, and other systems and application software in use for workstations as well as other platforms. CGA as well as other SI and professional services firms can not count on having the skills available that prospects and clients have in use or plan to utilize. Alliances, temporary staff members and the services of specialized temporary employment services must be used.

Skills bottlenecks are being encountered. CGA will propose changes in the software products planned or specified for jobs by users. Recently, one client was persuaded to step back to an earlier version of a workstation operating system in order to use the network manager which had been selected.

In addition to proposing changes to the set of software products selected for a project, the proactive step more often taken by CGA is to work with clients and recommend a set of products that can be well supported.

Internal education as well as relations with consultants and other firms are also sought to expand the skills inventory that CGA can support.

8. System Testing Process

More comprehensive tests are formulated with the aid of users, and users are brought into the review of proposed systems at an earlier stage through the use of front end CASE tools and JAD or RAD types of processes.

9. Particular Innovations

The effort to use the software engineering tools and concepts of leading software engineering firms together with the experience of CGA is felt to be innovative. Some other SI/professional services firms concentrate on using their own tools and methodologies. CGA wants to incorporate the ideas of firms who specialize in improving the software building process.

Position papers and internal training ensures that innovations are spread throughout CGA. CGA is prepared to work with clients to help them speed innovative changes in systems building, but this can be difficult to achieve in large organizations, since many user groups are acting independently.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

Improvements in systems building within CGA takes time and consume funds and personnel, and can delay getting work done and limit or preclude taking advantage of business opportunities. That is why CGA tries to use the experience of business partners active in systems development processes and tools as well as experiment with methods that might improve systems development. CGA wants to be able to work with clients that want to use various improvements, even if temporary consultants are employed to aid with that; however, CGA wants to limit internal changes unless there are longer term advantages. .

11. Success/Failures

CGA tries to avoid failures in using new systems building methods by working with business partners who are developing and testing out the methods.

12. Other Organizations to Watch

Leading systems integrators such as big 6 firms and CSC are always trying to improve their ability to build systems. Andersen Consulting stands out as a firm that they believe has achieved improvements.

13. Charging for Higher Value Software Assets

The total value of the assets can only be measured over time, although there may be more immediate benefits such as more rapid and less cost development or easier testing.

A systems integrator such as CGA can obtain recognition and remuneration based on the ability of delivering a solution. It might be necessary to establish a long term relationship to gain additional payment for delivering higher quality assets that would facilitate systems modification and growth for instance.

D. APPLETON COMPANY (DACOM)

1. Changes in the Software Building Process (Overall)

The current approaches, especially CASE, haven't just reached a dead end but have lead us down the wrong dead end: The problem isn't producing code efficiently and correctly, but designing the right systems in the first place.

DACOM began as a CASE-oriented, essentially product, company in the early 1980s. It built reasonably good tools and was (and is) a leader in repository thinking. However, over time DACOM saw that the chief need was for an understanding of business processes and making improvements in the systems which support business processes.

DACOM has evolved from a company with an emphasis on software building to one that works closely with clients to design the right system in the first place. Software "implementation" is viewed by DACOM as almost easy once the underlying business design is set.

Overall time to complete may or may not be shorter compared to conventional methods. Much depends on how well their clients have been educated in the DACOM method. Initial projects are usually slower because of education and cultural change. Subsequent projects are probably shorter; however, this is very difficult to measure since historical measures of time to complete don't do justice to the time spent on the up-front business engineering phase.

Actual software development is much faster after DACOM's work:

- There are relatively few downstream changes
- The "specs" are unambiguous

Quality is immeasurably higher, although, again, there are few benchmarks and measuring methodologies for quality. This is very frustrating for DACOM because they and their clients strongly believe this to be true, but it is very difficult to prove. Other vendors make similar claims -- which DACOM doubts are true -- and DACOM's real achievements (to them) get lost in the general noise level.

DACOM admits that they could do a much better job at quantifying benefits, in order to

- Measure their own progress
- Assess the relative strengths of different approaches
- Market themselves better

Right now, too much of their acceptance is based on clients buying into the DACOM "religion". This is ironic and frustrating for an organization that views itself as a rigorous, engineering-oriented company.

2. Soft Benefits

DACOM would take issue with the concept of "softer" benefits. The benefits are softer only from the standpoint that many (maybe most) of the benefits do not accrue to software building, but are gained by the underlying business. (DACOM agrees that it would be very powerful to measure the "before" and "after" business costs and benefits, but it does not do so on a consistent basis. Its clients don't want to pay for this kind of activity, because doing so won't help solve their particular business problem.}

DACOM works largely with true end users in the manufacturing/logistics area. From the standpoint of DACOM's clients the current system building process is "intolerable" and not just for the usual reasons of lateness and expense. The more thoughtful clients (and DACOM appeals to thoughtful clients, for better or worse) are frustrated by software that

- Doesn't meet their business functional needs very well.
- Behaves in inconsistent and unpredictable ways; users are overwhelmed by system details; computer systems make it harder rather than easier to understand underlying problems.
- Is difficult to change in the terms of time and expense and, even then, often can't really be changed to meet user needs

Downsizing of organizations and of the computing environment multiplies these problems

Their clients don't really care about the usual measures used to measure a "good" software project.

3. Relative Contribution of Tools and Processes

DACOM has a full perspective, having started as a CASE company (before the term was invented) and now describe themselves as a "systems integration consultant."

DACOM does use tools, some of which are their own (see attachment 1).

However, they see tools as essentially supporting a model-oriented approach ("Tools are the finger painting of business engineering").

- They see almost all CASE tools as not supporting the improvement of business processes (although pieces of them can do so with the right orientation)
- A good repository is vital, but they don't know of one that really meets their needs.
- They are forced to use and link standalone tools. They hope that in the future there will be integrated tools that will support a range of modeling techniques.

A key component (and bottleneck) is the need for consultants with the necessary "wisdom" to integrate the business problem, modeling techniques and client change. This is the strength of their business, but also a key limitation.

- These traits make them valuable to customers, but expensive.
- DACOM would like to be able to leverage this knowledge in a more embedded methodology. They see the application of AI and fuzzy logic as possibilities, but are currently resource-constrained in being able to pursue these directions very aggressively. (They are especially cautious since in the past they have tried to prematurely productize processes.)

The general process that they use is shown in Attachment 2.

4. Reusability

They are attracted to the concept of reusability but have found only a small number of "paradigm processes" which they can see as potential candidates. For example, "inventory" should qualify, but doesn't once the major differences between industries are examined as well as the "flavors" between firms.

They see no theory of reusability in the commercial world (unlike the formulas in the scientific world) to make their job easier. They use common skeletons and analyze differences but don't really view this as reusability.

They reuse "wisdom" and are aware of the deficiencies in doing so.

5. Difficulties in Addressing Client/Server

The client/server model is extremely incomplete. More basically, there is no model for organizational or functional downsizing. This puts a process engineering in a double bind. Learning for DACOM is "painful" and largely intuitive as far as where data, processing and location boundaries are drawn.

This disturbs them less than may appear, since the analytic process involved in understanding the business processes will usually keep them from making unrecoverable mistakes in "client" and "server" function allocations.

They feel great sympathy with their partners (whether customer-partners or vendor-partners) who are trying to make client/server technology work.

6. Is More Than One Process Model To Improve Systems Building?

They have their own approach (attachment 2). However, they feel agnostic toward particular business analysis practices and feel it is too early to back one horse. They have a high degree of control over the management of change.

7. Skill Needs and Distribution

Finding people who understand business processes is their greatest need. It takes several years to train a new employee with little business experience (largely on the job). They can predict fairly well the people who will not ultimately make it, after a day of pre-qualification testing; however, they can only predict those who will not fail; they are much less sure of those who will do very well. They have often been forced to hire more senior people as a result; this kind of person is more expensive, pound for pound, than a more junior person. A larger problem is that it is more difficult to form a cohesive "DACOM way" from people who are more set in their ways (even if these are sound, productive ways).

DACOM would like to emulate the philosophy of an EDS or Andersen and mold new recruits to their culture.

8. System Testing Process

Testing should be done at the model level.

9. Particular Innovations

DACOM believes that their entire approach is innovative, i.e.,

- Emphasizing modeling the upstream business process and defining objectives
- Focus on model building
- Making customer part of team, involving high levels of education and training
- De-emphasize conventional "software building" part of the equation (even though many of the DACOM staff has a heavy background in software building).

10. Cost of Making Improvements in Systems Building (Funds, People, Opportunities Foregone)

High costs, both to themselves and their customers. [A strength and weakness of DACOM is the fuzzy line between themselves and their customers.] DACOM has plowed back a great deal of time and money into iterations of their approach. They believe that they are on a very productive path now. But the whole process has definitely been a long term investment for them.

11. Success/Failures

There were several major failures in business concept and execution in the mid-1980s.

- They were overly product oriented
- They did not sufficiently value or leverage their "wisdom" in the past from a business standpoint

12. Other Organizations to Watch

James Martin Associates are doing good work. Otherwise, they admit to being inward-looking and do not "follow the competition". To some extent this is because they do not see most other vendors as being in the same kind of business as they are. That is, vendors that are working on software development improvements are not viewed as having many lessons for them. They view most other "business process re-eningeering" vendors as being much more touchy-feely (or management consulting oriented). But they admit that they may be wrong.

13. Charging for Higher Value Software Assets

They do not believe that their "consumers" are sufficiently educated to understand the issues involved. This goes back to the hard data question: If vendors (including themselves) could better demonstrate that this was true, then perhaps customers would pay more. They see this involving a long process of education and culture change -- 5 or 10 years at least -- before (a) This becomes true and (b) customers will accept it as being true.

Digital Equipment Corporation

1. Changes in the Software Building Process (Overall)

Changes are being considered or made to speed up the development of software as well as to reduce its cost and improve quality. Many separate activities are taking place ranging from steps in consulting and systems planning to work at all levels with tools and methodologies. Specific goals have not been quantified or can't be discussed, but it seems that the speedup being sought at this time would reduce development time by one third on larger projects.

2. Soft Benefits

There is a definite need to achieve other, softer benefits for customers including improved satisfaction with the development process, improved communications on the work being done, and more ability to respond to needs. These needs are being investigated in relation to systems integration and professional services work as well as in relation to internal analyses of systems building.

3. Relative Contribution of Tools and Processes

Tools including CASE tools and other software aids are being sought and evaluated to aid with the system building process. Tools may seem to be more important in regard to current improvements in systems building, but improvement of the entire process that is carried out in systems building offers more opportunities for improvement.

The use of techniques such as JAD or RAD as part of a process are expected to have more impact on time, cost and quality than tools can.

Research is being conducted that addresses both the use of tools and processes to improve the system building process. The use of manufacturing techniques in software production has been carried out and is also being investigated together with other topics.

4. Reusability

Tools are being used that allow a repeatable solution to be sold. For certain applications, these tools will allow the solution to be modified or customized more easily, and many components of the solution would not have to be changed. There have been various types of reusability in the past by DEC VARs as well as by DEC. Digital has also been experimenting with reusing objects in a group of internal projects and is now doing that on a regular basis

5. Difficulties in Addressing Client/Server

There are questions about the allocation of data and processes from a consolidated system to a number of client/server systems. There are also questions about using a set of data or transactions received at one point to update multiple client/server systems.

6. Is More Than One Process Model to Improve Systems Building?

DEC is using multiple process models, but most of these efforts are coordinated. In some circumstances, activities in support of accounts could use tools and techniques or new methodologies in an unplanned manner to improve systems building. Where a VAR or associated vendor is in a lead position, this is particularly true. They have a medium degree o control over the management of change.

Tools are available to help manage or control change management at DEC, but they are not always used.

7. Skill Needs and Distribution

There are a number of new skills that are needed in current projects ranging from those needed to use new systems development and relational database tools to those available to use network software products, GUI and workstation spreadsheet and database products.

The new skills are not always present at DEC. Some internal training is carried on and external training is made use of. However, a group of contract services vendors are used to meet critical situations with temporary people. VARs and associated consultants are also used to meet critical needs.

8. System Testing Process

Efforts are being made at Digital to develop new types of testing that will be automatically incorporated into the system building process.

There are also some products being introduced jointly with large companies that will handle portions of the testing process in new ways through system review with front end CASE tools, prototyping and other means.

9. Particular Innovations

Digital feels that its system building process is innovative since it results in complex network solutions in a relatively short time.

Several recent internal efforts have resulted in factory-like development of code, but can't be discussed yet.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

It is hoped that there will be some rapid or near term improvements although most observers at DEC feel that it will be a long term investment.

11. Success/Failures

There have been difficulties encountered, but failures have not been recorded or won't be discussed.

12. Other Organizations to Watch

These range from organizations concerned with improvements such as James Martin & Associates and KnowledgeWare to vendors such as Andersen Consulting and Cambridge Technology. Andersen has a methodology and tools that have been demonstrated on a number of projects.

13. Charging for Higher Value Software Assets

Not sure how it can be measured and recognized, except over a period of time.

It will be necessary to guaranty the higher value in some way, if users are asked to pay more.

Hewlett-Packard

1. Changes in the Software Building Process (Overall)

The driving force behind changes in the systems building process at HP is to reduce the time required to deliver projects while improving the overall quality of the finished product. HP's area of emphasis is on the management process for building systems, not on tools. They are placing emphasis on the re-engineering aspects of the business process, and simultaneously trying to develop a methodology which will apply a rigorous discipline to the resulting systems implementation.

They do believe that tools are important. However, they believe that 90% of the errors occurred at the front end, where few tools are of much use without an overall re-engineering of the business process.

They believe that their emphasis on re-engineering focusing on downsizing, organizational changes, process re-design within reasonable cost containment guidelines of the client, is the primary thrust of their rethinking of the systems development process.

Reduced cycle time on the completion of business systems re-engineering project, including the generation of finished applications systems - 25%. They would not give a specific example.

HP is trying to define quality as "fit to business need" within the client's anticipated time frame. Their position is that counting bugs in the code is secondary.

HP has not made much progress on reusability. Reusability per se is not a goal. Studies that they've conducted indicate that there is huge waste in the current process which could be significantly reduced if a set of standards could be developed along with a process to better manage code; identify and select modules to be developed and maintained as "reusable", etc. However, they feel that the process is a long way off.

The main problem is that in order to generate reusable modules, the development process must be integrated or contain components and standards which insure reusability. They think that most firms are farther behind in this area than they are in CASE. (In HP's view, most firms aren't very far along in the effective integration of CASE into the development process.)

Very little is being done to obtain any more meaningful quantification of benefits than was done in the past. This process becomes even more complex when significant business re-engineering is involved. In some instances various components could be readily quantified, but that inmost instances where business investment on the basis of much softer criteria.

2. Soft Benefits

HP's experience has been that the two dominant soft benefits that firms are using to justify re-systemization (and business re-engineering) efforts are:

- Improved customer satisfaction; and
- Reduced cycle time on core business transactions

Certain of HP's customers have taken to trying to measure both of these factors; but it's still difficult to relate the measures to the bottom line performance of the firm.

3. Relative Contribution of Tools and Processes

HP believes that the biggest contributions are coming through improvements in the process and people management aspects of the systems building process. CASE tools and other automated approaches are important but secondary. They have adopted an approach which pairs top-flight project managers with tailored teams for specific implementations. Project managers are responsible for projects only. Team members are managed in the personnel and development sense by other managers within their home organizations.

- They have installed new processes that permit feedback between project managers and those managers in the firm who are responsible for the development and administration of individuals within the organization.
- They have installed a special program to continually improve even seasoned project managers skills.
- They are using some incentive programs for project performance and quality on an experimental basis, but these programs are immature in that they tend to reward heroic fire fighting efforts...rather than solid fire prevention on a day to day basis. Incentives (both team and individual will probably play a stronger role in the future.

4. Reusability

Very little is being done by HP in the reusability concept as applied to custom code development. However, a significant amount of work is going into refining process which will insure reusability of some significant portion of the code that's developed for applications or systems software products.

What is being done in this area is proprietary to HP and its products. They are still a long way from the point where the generation of "universal objects", etc. will be commonplace

5. Difficulties in Addressing Client/Server

HP has been working at the C/S model in an open systems environment for some number of years. They are very comfortable with it at this point, and have a formally documented approach for allocating functions and data between clients and servers. (As is the case with reusability, the model is proprietary to HP and might not be applicable to other platform architectures.) The model is supposed to be rigorous and has significantly improved the quality and efficiency with which new applications can be developing using C/S.

They still needed to improve the overall process at the design level. The area that needs the greatest improvement is a consistent approach to linking redesigned business processes with functional specifications for C/S systems. They are working on this.

- They have just dropped KnowledgeWare in favor of IEF as their primary CASE vendor.
- Anticipate that IEF organization will work jointly with them on tailoring CASE tools to meet HP's needs both at the design and implementation levels.

A formal model for the C/S design process has significantly improved the quality and reduced the time required to systematically test new code. Estimates that they are saving about 20-25% in this area, and are certain that the quality of the code is higher.

6. Is More Than One Process Model to Improve Systems Building?

HP is definitely trying multiple approaches to improving the systems building process, but the process is not seen internally as very well coordinated. Individual initiatives appear to pop up, and get piloted. Informal communications between people involved in these efforts are the primary method of coordination. However, enough information gets shared to insure that some "best of breed" approaches are emerging.

There is a group manager whose responsibility it is to gather, digest and redistribute information on the new techniques and approaches that emerge. Overall, they have a medium to low degree of control over the management of change.

7. Skill Needs and Distribution

Technical skills don't seem to be a problem for HP. The company's culture supports investing in employees, particularly in technical education. Project management is the primary area where they see a significant shortage. And, there are no magic wands to wave that eliminate the need for the "experience" factor in project management. They are trying an "understudy" approach which pairs less experienced project managers with more experienced ones to hopefully obtain knowledge transfer.

8. System Testing Process

Adoption of the C/S model has significantly changed (and improved) the testing process. (See the answer to question 5.) Since the modes provides for a limited set of structured communications between client and server using standard protocols, it's possible to develop test packages to automate the testing of each units functionality separately and in most instances simultaneously. This has lowered testing costs and cycle times significantly.

9. Particular Innovations

They believe that there is significant variation between major firms, but that probably all leading firms have hot spots where advances are being made. Several user companies and two vendors were cited.

- Dupont - Significant work in "rapid" prototyping for C/S applications
- Farmers' Insurance - Advanced architecture for C/S transaction management
- Microsoft - Use of advanced tools and concepts (object oriented coding, etc.) to automate the systems building process. Reusability at applications product level.
- HP - C/S applications model and transaction management
- Andersen - Re-engineering process and (probably) reusable applications modules.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

The following is HP's educated guess:

- In addition to whatever investment cost their might be in designing a new process or technique, the first time the process is put into use on an actual systems effort, there is probably a twenty percent penalty in development cost.
- On the second attempt the penalty probably drops to 10% and on future efforts gains in productivity can be made in the order of 10-20% per effort.

To pull together HP's C/S model probably cost between $500K and $1M. This doesn't count the many hours that were invested in developing stand alone components which were used in the model. However, benefits are there from the developers viewpoint.

After two years of working with the model in actual development efforts, they are probably delivering systems at 80% of the cost for these same systems developed using conventional methods.

11. Success/Failures

The C/S development model has been Hp's biggest success. On the failure side, early on they had an occasional disaster in terms of actually using the model to do a client implementation.

12. Other Organizations to Watch

HP is probably ahead of the average firm in rethinking and actually implementing changes to the software building process. They believe they were probably far from the head of the pack. They don't think that there was any one firm who was the leader in all aspects of the process.

13. Charging for Higher Value Software Assets

HP doesn't feel that the asset value of individual custom designed systems will improve significantly as the direct result of improvements to the process; but does believe that the process will create new kinds of assets.

The business process models developed as a result of re-engineering can and (in a few instances) are being sold or brokered. Repositories of basic business processes will one day be sold. Using these models through a medium such as IEF, other firms could tailor both the processes and the code to meet their particular business needs. One example cited was an airline used TI to broker its IEF model for frequent flyer operations to another airline serving a non-competitive market.

IBM Corporation

1. Changes in the Software Building Process (Overall)

There are changes being made by groups in a number of areas including professional services, external consulting, SI projects, ISSC and internal software groups. IBM's interests include business systems planning, project management, methodologies and tools. Separate activities are also being carried out with business partners.

Internal professional services and software groups are under pressure from users to speed up the development process or time to develop an application system. The reduction of development time by 50 to 100% is mentioned by users as an objective although any meaningful reduction in the overall process (e.g., 20 - 33%) would be well received. IBM recognizes these objectives, but does not have a specific target. However, IBM recognizes that changes of this magnitude will require consideration of all aspects of the process from concepts of business analysis and re-engineering to project management and code generation.

Users also want to reduce the cost of developing software for a system. The reductions being sought by users are at least 30 to 40% and as much as 100% Although there is not any official target, IBM research in Endicott, NY has talked about helping users achieve a reduction of 25 to 30%.

Another objective is to allow the time and cost required to modify software to be reduced.

A different type of objective is to develop software with a higher level of quality. In addition to programs that can be more easily modified and understood, two concepts were mentioned in regard to this. Users have stated that they are interested in systems building that would automatically incorporate various types of input data validation and other cross footing that is suitable for the type of application being developed. The second concept involves enriching the development process or adding tools which will check out the logic of software during the process and leave a structure in place for acceptance testing and testing of modifications at a later date. Auditing should also be facilitated.

Many ideas about quality are prevalent. One recommendation mentioned by users is to ensure that developed software should not be fragile or become fragile when normal types of changes are incorporated or added. Users also feel that procedures or tools should guaranty that software modules can be reused or used as an object(s).

2. Soft Benefits

Software should facilitate efforts to communicate as well as to analyze what is going on in business processes. There should be a view(s) of software that enable users to review systems and give them a feeling of satisfaction with the software.

3. Relative Contribution of Tools and Processes

CASE won't achieve all the objectives that are being discussed for software but CASE tools can aid by enabling software to be analyzed, modified and generated more easily for a number of applications. It was noted that certain applications that require high levels of transaction processing might require attention from personnel that were experts in the alternatives and tools available to meet tasks. One project mentioned in this regard involved an RS/6000 system that AIC worked on in conjunction with the research staff in Austin.

An expert on GUI has been hired by IBM in Endicott, NY to develop special tools to support and guide developers in implementing certain types of workstation systems faster and with more quality.

In regard to the use of tools, it is felt that AI and particularly expert systems, as well as other capabilities should be able to be incorporated as objects in the software development process.

It is felt that the process of systems development must address a much broader set of needs than tools can address. For instance, it must address changing work structures such as work groups and other changes in work organization or people management.

Ideas for the application of manufacturing techniques to software building are also being considered, tracked and experimented with although it is felt that it may be difficult to achieve other objectives such as improvements in communication as mentioned above together with software manufacturing.

4. Reusability

Reusability was mentioned several times as an objective in relation to the goals stated above. The objective was stated as creating software that can be reused and/or facilitate the reuse of other software whenever feasible.

Ideas have been tried out internally by IBM in Boca Raton, Austin, and other locations to reuse software during systems software development.

Research projects have also been carried out to test reusability and other concepts and to track user and vendor work with reusability.

5. Difficulties in Addressing Client/Server

Difficulties involved in separating processes or splitting the management of data have not been fully addressed. IBM thinks that these problems have often led to the use of larger capabilities, AS/400s rather than work stations, in many client/server systems.

6. Is More Than One Process Model to Improve Systems Building?

IBM has a number of process models available to improve systems building and is constantly in contact with other vendors to make arrangements to share and try out their ideas (e.g., recent agreement with HP on client/server development products). They have a low degree of control over the management of change. Many offices (but not all) are supportive of change due to this. IBM contacts in professional services assignments feel that this emphasizes that the most important element in the improvement of systems building is the experience and knowledge of industries and application systems.

IBM points out that some large corporations such as several of the top banks are trying to use more than one process model for improving systems building. This has occurred chiefly due to unplanned pressures by users for different alternatives for systems building. There may not be an adequate means of managing or coordinating change in these situations even if IS tries to coordinate or manage change in situations where more than one model is operative.

Some areas of organizations are more supportive of change, particularly users who are under pressure to achieve business goals.

7. Skill Needs and Distribution

Skills are still in short supply and distributed unevenly. Hiring restrictions and staff reductions have made it difficult to get the right people. The IBM culture still places great emphasis on training. Proactive steps to improve the situation include arrangements with vendors that can aid with or participate in improvement in software building.

8. System Testing Process

Users are being brought in to systems testing to a greater extent. New types of tools or systems review are being sought to improve the ability of users to test systems. Where vendors are involved in the supply of or development of software products, more accountability is being placed upon them during systems testing.

9. Particular Innovations

Innovation of various types is being tried by corporations in contact with IBM including the development of specialized tools internally or by vendors, the generation of objects in C++ code that can be used to add necessary functions to a number of systems, the use of specialized CASE approaches, greater reliance on vendor application software products and experimentation with new types of languages. One effort that IBM is engage in with a user involves the use of specialized GUI that aid a developer to combine objects. Use of the GUI will guide (and force) the developer to include various types of controls and aids that will facilitate testing and operation of the resulting system.

IBM is also conducting research on ideas about GUI that use symbols other than icons. One of these concerns the use of symbols that would aid a person involved in a work group structure.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

The cost of making improvements can not be tracked on an overall basis in IBM and is not being tracked in most organizations. IBM is willing to make improvements to obtain near term business or to improve capabilities over a longer term. It is felt that many short term efforts should be undertaken in order to ensure that IBM will be aware of the more outstanding improvements.

11. Success/Failures

Failures or the inability to achieve total objectives has been notable in some CASE projects as well as projects run by IBM that were supposed to improve software building. A major lesson that was learned was that the business objectives and processes are sometimes neglected or not reviewed sufficiently and rethought during projects that automate systems building.

12. Other Organizations to Watch

The vendor mentioned most often as one who is or may be achieving such improvements is Microsoft. They have been building and rebuilding software objects so that they can be joined together more easily. Visible Basic has been improved and re-coded in C++ to provide users the means of reviewing and adding to or modifying objects in a more easy fashion. The Next operating system of Jobs was felt to be a good step in the direction of improved development, but is not spoken of lately. Products such as Powerbuilder and the Insynch Passport system that generates C code are also being reviewed in addition to the products of TI, KnowledgeWare, Bachman and others whose products are under constant review.

13. Charging for Higher Value Software Assets

Although it is believed that higher value software assets will be developed, there are unanswered questions about the ability of a vendor to achieve sufficient recognition for the software assets put in place so that a premium can be charged on a near term basis.

Oracle

1. Changes in the Software Building Process (Overall)

The principal changes taking place in the process are driven by three fundamental driving forces.

- A migration from IS to the end-user as the leading player in the development process.
- The maturation of a new life cycle based on rapid prototyping, utilizing increasingly sophisticated tool sets.
- A growing demand for rapid turnaround and responsiveness for changes to whatever system is in place.

Both downsizing from the mainframe and upsizing from standalone departmental systems to departmental work group systems is occurring simultaneously. In either case, the traditional life cycle that was used for mainframe development is or will be replaced by the cycle described below.

- Rapid prototyping of a systems basic functionality. (Joint effort of specialists - either vendors or IS - and end-user experts.) The prototype then goes into limited production.
- Refinement and rollout of the full blown system. (Joint effort of specialists and end-user experts, and based on feedback from operating the prototype in production mode.)
- Integration of the refined system with related systems. (Primarily a specialist implementation activity based on requirements identified while operating the refined system in production.)

In other words the traditional concept of a formal requirements phase is replaced by three less formal requirements definition phases. In fact each phase of the traditional life cycle has a three counterparts in the new cycle; design, development, testing, etc.

Oracle sees itself as being involved in this process in several ways:

- As an innovative tool supplier
- As a teacher/coach
- As a high level consultant and/or implementor
- As a developer of applications software (for internal use or, perhaps, in partnership with a client)
- As a supplier of flexible applications (where Oracle may again become involved as a teacher, consultant or implementor)

These are complex roles and Oracle is the first to admit that everything isn't sorted out yet.

Meeting the real business requirements of the end-user, and shortening the time from inception to delivery in order to capture user benefits early. This fundamentally comes down to:

- Reduced training time
- Shortened development cycle to first usable version of the system
- User self-sufficiency in terms of on-going modifications, etc.

Oracle does not believe that there is likely to be a reduction in systems development costs (at least in terms of reduced manpower) over time. The more likely scenario is that implementations will have four to five times the functional capability for the same cost.

Oracle believes that the creation of new interactive applications using their redefined life cycle and tools can be accomplished in 70% of the time it would take to achieve the same functionality using conventional CASE and methods. For batch based applications or the generation of new reports, etc., they believe the number is 25%. (See attached diagram showing how newer approaches to system building generate benefits.)

Quality is customer satisfaction with the provided functionality, and "0 defect code".

In a homogeneous environment (Oracle Platform) they estimate that a fair amount of the code can be re-used. But this is not an objective unto itself. In heterogeneous environments (say using DB2 along with Oracle), less will be achieved in terms of reusability.

Oracle believes that the industry is a long way off from obtaining reusability on any major scale through the development of "objects". Probably the highest level of reusability comes through the creation of user "customizable" applications software products.

Very little quantification of benefits has been done. Most end-user projects (which are an increasing proportion of the total) seldom undergo a formal justification or subsequent audit process.

2. Soft Benefits

As firms begin to leverage the concept of information distribution and collection utilizing the interactive capabilities of modern technology significant benefits will be realized in terms of better communications both deferred and direct. This will include interaction between individuals in a firm as well as between an individual and the corporate repository of knowledge. Applying these same systems design concepts to interactions with customers, etc. will bring another whole set of benefits.

The typical knowledge worker spends between 10-15% of his/her time today interacting with paper. As the level of integration of systems goes up and the ability to interact on an as needed basis with the corporate repository grows, knowledge workers will spend probably 20-25% of their time working with information electronically.

In Oracle's view we're just seeing the tip of the iceberg in terms of understanding, articulating and realizing some of the significant benefits that can be achieved through the migration to interactive systems at all levels.

3. Relative Contribution of Tools and Processes

New tools and changes in processes are probably making an equal contribution; and they continue to evolve in an iterative manner. The methodology that Oracle endorses of rapid prototyping, refinement and integration has stimulated the need for new functionality in the tool set. And, the capabilities that are subsequently generated in the tool set provide opportunities for further refinement in the design and implementation processes.

CASE tools are either obsolete or essential. If they are integrated with rapid prototyping methodology and deal with the concept of distributed architecture, they are essential. Tools that "assume" the kinds of structured methodologies of the 70's and 80's are probably obsolete because they don't integrate with current methodologies or architectures.

AI, like many other buzz words (including CASE), has gotten a bad name. However, if you look around, you'll see that expert rule based systems are imbedded in everything from spreadsheets and word processors to systems development tools. The use of these technologies to support the systems building process will continue to grow, and will result in both reduced development time and higher quality code.

The biggest single change on the people and organization side was the migration of development responsibility from IS to the end-user. Distributed IS has changed the process. The result is more teamed approaches which hopefully combine the required technological expertise with user knowledge of business process.

In terms of management techniques, people are still going to be people and managers will still vary in quality. However, the teaming of users and technical personnel in the development process will and should continue to reduce the complexities of management by improving communications. New tools and processes will continue to reduce the number of people required to accomplish an implementation (and therefore the complexity).

Oracle believes there is some fallacy in the notion of "manufacturing" code. Manufacturing implies making a large quantity of the same (or essentially the same) thing. Building applications involves using common processes to generate unique systems. The thing that they have in common is the use of consistent (but obviously different) processes... little else.

4. Reusability

(See Question #1)

From Oracle's point of view, reusability is an objective. However, a lot of effort will need to be placed into the development of standards, and the consistent use of sophisticated tools before this will be accomplished through "object" libraries. In the meantime, Oracle targets on being able to reuse higher level application definitions to "regenerate" code. Modifications to the definitions at the high end allow customization; and the use of tools allows new custom code to be generated at minimal cost.

There may not be much progress was likely to get made in generating reusable modules at the user interface, since this is an area where systems providing similar functionality were liable to vary significantly based on personal and organizational preferences.

5. Difficulties in Addressing Client/Server

Oracle believes that the C/S model is incomplete. When utilizing a consistent platform on any given set of applications (the Oracle platform) they have a consistent way to model the distribution of processing and data. It's pretty straightforward. Shared functions and data go to the server, and unique functions and data go to the client. However, when working with heterogeneous platforms (in particular data bases), the ability to follow the model is frequently inhibited by differences in functionality in hardware and software.

Over time, the migration to more open systems and standard interfaces will likely increase the level of consistency that can be achieved in allocating data and processing in the C/S environment. In any event, its not a serious enough problem that it is likely to inhibit the ongoing emergence of C/S as the primary applications platform.

6. Is More Than One Process Model to Improve Systems Building?

Yes, Oracle is certainly trying more than one model. Two examples have already been cited; the rapid prototyping process and the creation of "customizable" applications packages. More are likely to be tried in the future.

In terms of how the process is being managed... at Oracle it's ad hoc. Overall, they have a low degree of control over the management of change.

7. Skill Needs and Distribution

Oracle, being a technology company, makes heavy investments in training, and provides numerous incentives for self-development. On-going training and development is just part of the culture. So fundamentally, Oracle doesn't feel that it's facing a any significant problem in finding or developing the skills that it needs.

In general, Oracle sees the distribution of IS skills to user organizations is a significant change, and will continue to happen throughout the decade. In terms of skills bottlenecks, Oracle believes that these are largely transient effects.

8. System Testing Process

Testing is becoming an on-going continuous process. The objective is no longer just to meet a written specification, but to insure that the customer is satisfied with the cost, functional performance and time to delivery of the system. In a sense this increases the complexity of the entire testing process, and places the "pass fail" judgment directly to the user.

In terms of actually testing finished code, major tests are built in at each of the three major phases of the Oracle process (rapid prototyping, refinement, integration).

The process of regression testing of software is growing more complex. As more and more functionality is included, the possibilities that need to be tested multiply exponentially. Some automated processes have been put in place, but manual testing of individual products is still required.

9. Particular Innovations

- More and more integration of applications systems and data
- Installing methodologies which permit the continuous refinement of systems
- Providing tools and methodologies which allow users to achieve high levels of self-sufficiency in terms of managing their own systems environment.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

Unfortunately this is something that isn't measured. Partly because what is being done is intertwined with so many other activities, it's not clear that even a heavy investment in the measurement process would produce a result of significant value. However, major investments in new methodologies and tools were looked upon by users as long term investments. Only when the first applications came rolling out under the new approach would people begin to realize the potential savings or pay back.

11. Success/Failures

It's no secret that Oracle's biggest failure was in its initial approach to the applications software market. The products (originally developed for internal use) were low in quality and functionality, support was low quality, and sales commitments exceeded the deliverable capability. This problem is behind them now, and caused a major rethinking about the entire applications building process which forced Oracle into its new direction - with benefits all around.

12. Other Organizations to Watch

Some significant improvements are being made in the pharmaceutical industry. (Examples which confidential.) There are similar developments in the chemical where the adoption of new techniques and approaches to systems is allowing high levels of integration improving overall productivity to the firm.

13. Charging for Higher Value Software Assets

The actual asset value of software will probably go down as it becomes easier to build and is replaced more quickly. However, the value of the "information asset" will likely rise significantly as newer applications and software systems permit its distribution, collection and utilization in ways that weren't previously possible. Whether anyone can measure these values is debatable.

JULY 1993

Assessing Trends in the System Building Process of Selected Vendors

The Atrium at Glenpointe, 400 Frank W. Burr Boulevard, Teaneck, NJ 07666 201-801-0050

ASSESSING TRENDS IN THE SYSTEM BUILDING PROCESS OF SELECTED VENDORS

I. BACKGROUND AND METHODOLOGY

A. Background

Andersen Consulting is considering making changes to the way in which its software solutions are created and delivered. To assist in this process Andersen engaged INPUT to examine what other vendors are doing or planning to do in this area.

Andersen and INPUT staff met in Chicago to review the background of the study on May 6, 1993. Over the next week the vendors to be targeted and the issues to be addressed were finalized.

B. Methodology

An interview guide was developed and reviewed by Andersen. The guide was intended to be a departure point for understanding the other vendor's views and plans across the broad area of system building. A copy of the Interview Guide follows the end of this chapter. The basic topics covered are summarized in exhibit I-1.

The vendors in exhibit I-2 were targeted for interviews. All vendors were interviewed. Because of the lateness of the interview, the analysis of TI is not contained in this report.

INPUT made in excess of 100 contacts at these firms to obtain information.

- In some cases, a single person is able to provide all or most of the information required. This is generally because the company has centralized the function and/or has it well-coordinated. Even in this category, it has often taken a number of tries and referrals to find the correct person.

- In other cases, responsibilities for software building are diffused throughout the organization. Up to six interviews have been required in this type of organization.

Exhibit I-3 gives examples of the titles of primary suppliers of information.

INPUT was not sure at the beginning of the study what kind of reception the study would receive. In fact, every company approached has been very interested in the subject and, equally important, quite willing to take part in the study. The interview guide itself has been a significant "selling point" in motivating vendors to take part. The only limitations have been finding the right person and accommodating their schedule.

Because of this elevated level of interest INPUT has not found it necessary so far to promise anonymity. In a few cases proprietary or sensitive information has not been reported here; but this kind of information has been infrequent and, in INPUT's opinion, has no effect on findings.

INPUT believes that being able to associate the findings with actual companies, rather than, "A", "B", "C", etc. will make this information more useful for Andersen. INPUT does request Andersen, however, that this information be treated with care and not widely publicized, at least without masking the identities of the vendors adequately.

C. Organization of This Report

The remainder of this report is organized as follows:

Chapter II: Summary Analyses - These are INPUT's conclusions

Chapter III: Summary Matrixes - This very concisely summarizes material by vendor under the 13 categories in the Interview Guide. For more detail, the full text in Chapter IV should be consulted.

Chapter IV: Summary by Category - This provides the full text for each vendor within each category.

Chapter V: Summary by Vendor - This is the same information as in the preceding chapter, but organized so as to give a profile of each vendor.

Exhibit I-1

TOPICS COVERED IN VENDOR INTERVIEWS

1. Changes in the Software Building Process (Overall)
   - Techniques
   - Development Time
   - Cost Reduction
   - Quality
2. Soft Benefits
3. Relative Contribution of Tools and Processes
4. Reusability
5. Client/Server Model
   - Problems Seen
   - Vendor Direction
6. System Building Models/Approaches
7. Skills Needed and Distribution
8. System Testing Process
9. Innovations (Self-Assessed)
10. Cost of Making Improvements in System Building
11. Success/Failures
12. Other Organizations To Watch
13. Charging for Higher Value Software Assets

Exhibit I-2

Companies Interviewed

Borland

CAP Gemini

Cambridge Technology

D. Appleton (DACOM)

DEC

EDS

HP

IBM

Microsoft

Oracle

Texas Instruments*

Xerox

* Not Analyzed in Report

Exhibit I-3

Examples of Primary Respondent Titles

Executive Vice President

Vice President, Software Development

Director, Solutions Strategy

Director, Worldwide Technical Marketing

Director, Systems and Methods

Process Engineering Group Manager

Director, Professional Services Research

Senior Software Research Analyst

Product Development and Tool Manager

Senior Research Scientist

Director, Technology and Methodology

II. SUMMARY AND OBSERVATIONS

This chapter briefly summarizes the detail reported in the remainder of the report. This is the only section of the report where INPUT makes some of its own observations; these are labelled as such.

The first section of this chapter reports macro findings. The second section summarizes detailed findings.

A. Overall Conclusions

There is great diversity in the approaches being taken by the vendors interviewed: No two are following exactly the same path. INPUT believes this diversity exists for several reasons:

- There is in fact no consensus yet as to a "best in class" approach to software building. All companies are, to a certain extent, having to feel their way toward the desired objective.
- Objectives themselves differ also: Is a firm primarily building software for others or for itself?
- Company culture matters a lot also: What works for an Oracle or HP may not work as well for an IBM or an EDS (or vice versa).
- Finally, it may very well turn out that there may be quite different software building processes that work equally well, or that work equally well under different sets of circumstances.

However, within this diversity, there are also some commonalities. The main areas of commonality include:

- A focus on process as opposed to tools. INPUT believes that this emphasis is largely correct, however, INPUT would make these additional observations:
  - A few years ago, tools may have been oversold; now the reaction has set in.
  - Existing tools have not caught up to client/server needs and specific client/server tools are still in an early stage.
  - INPUT's belief is that tools and process will ultimately be balanced.
- JAD and RAD are commonly used.
- Object-driven development is less frequently cited, but may turn out to be quite important.

- The people skills required are commonly seen as changing rapidly; this is consistent with JAD and a focus on user requirements.

  - Obviously, training is often cited as a way of reskilling personnel. However, some companies have noted that trying to change the focus of an essentially technical person is not always feasible.

  - There is another issue which surfaced which INPUT believes that all development organizations should be cognizant of: The "return of the super programmer" (or the "super user interface").

    There may only be a limited number of people who have the skills and attitude to function well in a JAD, applications-driven, client/server environment.

- The client/server environment does not seem to be well understood. Most companies agree that the client/server model is not complete or well-understood. While the thought was not always vocalized, there is a danger of creating (or trying to find) a new generation of software artists.

All firms interviewed are very active in trying to improve their software building capabilities. They believe that time to complete and quality will be improved, in some case markedly improved. Companies are much less sure whether there will be sizable or measurable cost benefits. Some companies are already reporting significant improvements in customer satisfaction, which they believe can be traced back to improvements in software building.

B. Detailed Findings

This section, and its associated exhibits, summarizes the findings under each of the major questions, or functional issues, posed in the interviews. This summary information ties back to the exhibits in Chapter III. (Chapter III also ties the detail to individual companies interviewed.)

Changes in Software Building Techniques (Exhibit II-1)

JAD and RAD are the techniques most often used.

Changes Seen in Development Time (Exhibit II-2)

Every vendor sees a reduction in development time due to changes in the software building process. The median improvement seen is 30%. Two see reductions by over two-thirds.

Cost Reduction (Exhibit II-3)

Most companies see the changes reducing costs. However, very few can quantify these reductions. [The inability to quantify savings means, in INPUT's opinion, that these claims should be treated cautiously.]

Improvements in Quality (Exhibit II-4)

All firms see quality improving, but find it difficult to quantify the change.

"Soft Benefits" from Changes in the Software Building Process (Exhibit II-5)

The benefits most often cited are improved customer satisfaction, business re-engineering and improved human communications. There is some linkage and overlap between these benefits.

[Re-engineering benefits are probably quantifiable, but are often outside the scope of "software".]

Relative Contribution fo Tools and Processes (Exhibit II-6)

Process is seen as important by all but one vendor. [See further comments in Section A of this chapter.]

Reusability (Exhibit II-7)

Many vendors are still assessing reusability. INPUT believes that this will become an increasingly important issue. The ones that already see it as important generally associate it with object-oriented development.

The Client/Server Model (Exhibit II-8)

Most vendors see the model as incomplete and are trying to address the issues, especially on how to divide processing and data. Interestingly, a minority see the incompleteness but do not view it as a critical issue.

System Building Models/Approaches (Exhibit II-9)

The majority of vendors have not yet settled on a single model. Most probably expect to. There is a wide degree of variance on how much management control or coordination there is over the approach to systems building. A low degree of control is not necessarily associated with ineffectiveness now (due to the state of flux the entire discipline is in); however, this could be a problem in the future.

Skills Needed and Distribution (Exhibit II-10)

Skills needed are generally seen as changing and, therefore, a bottleneck. [See further comments in section A of this chapter.]

System Testing Process (Exhibit II-11)

The major change is toward early and/or parallel testing.

Vendor Self-Assessment of Their Own Innovations
(Exhibit II-12)

There is little overlap on what each vendor sees as innovative areas. This is consistent with the diversity of approaches.

Cost of Improvements to System Building (Exhibit II-13)

While not generally known, costs are seen as significant. Both long and short term benefits are cited.

Success and Failure (Exhibit II-14)

In some cases, prior failures led to more attention being paid to the improvements in the software building process. Respondents did believe that the changes in the process have led to success, but had difficulty in making concrete connections.

Other Organizations to Watch (Exhibit II-15)

Andersen was cited most often, followed by Microsoft and TI. The level of actual knowledge as to what other firms were doing was relatively low.

Charging for Higher Value Software Assets (Exhibit II-16)

Most firms believed (or would like to believe) that software could be a higher value asset. However, most firms had trouble with how customers could be convinced that this was so or, even, how the value could be computed.

Exhibit II-1

Changes in the Software Building Process: Techniques

- JAD (5)
- RAD (5)
- Object (3)
- Other
  - Client/Server
  - Function Points
  - Modeling
  - Time Boxing

Exhibit II-2

Changes in the Software Building Process: Development Time

- All see Reductions
- Nine named target percentage reductions in time:

  20%
  20%
  25%
  30%
  30%
  30%
  33%
  67%
  75%

Exhibit II-3

Changes in the Software Building Process: Cost Reduction

- Nine of 11 see cost reduction
- Cost reduction targets less certain
  - Somewhat (3)
  - Probably
  - Especially maintenance
  - Significant
  - Percentages cited:

    20%

    25 to 30%

    50% (maintenance)

Exhibit II-4

Changes in the Software Building Process: Quality

- All see quality increasing; relatively few specific targets
- Zero defects; 30 to 50% fewer bugs

Exhibit II-5

Soft Benefits

- Customer Satisfaction (5)
- Re-engineering (5)
- Communications (human) (5)
- Maintenance (2)
- Cycle Time (2)
- Improved Specifications (2)

Exhibit II-6

Relative Contribution of Tools and Processes

- Process is primary (8)
- Tools are primary (1)
- Equal (2)

Exhibit II-7

Reusability

- Important (4)
- Assessing (5)
- Not Important (2)

Exhibit II-8

Client/Server Model

- Incomplete - Being Addressed (8)
- Incomplete, but not a critical issue (3)

Exhibit II-9

System Building Models/Approaches

- Single Model Used (4)
- Multiple Models Used (7)
- INPUT's Assessment of Degree of Management Control Over Process
  - High degree of control (3)
  - Medium/high control (4)
  - Medium control (1)
  - Medium/low control (1)
  - Low control (2)

Exhibit II-10

Skills Needed and Distribution

- Skills widely seen as bottleneck (8)
  - Training seen as solution
- May be a difference between vendors who are "people dependent" (e.g., DACOM, Xerox and Borland) and "process dependent" (e.g., EDS)
- Three vendors are less concerned over skills (HP, Oracle, Microsoft)

Exhibit II-11

System Testing Process

- Nine vendors plan changes
  - Seven plan more early/parallel testing
- Two do not see testing as a priority

Exhibit II-12

Innovations (Self-Assessed)

- Little commonality between innovations
- Examples:
  - Methodology
  - Object/Reusability
  - Modeling
  - Software Factory
  - Code Generation

Exhibit II-13

Cost of Making Improvements in System Building

- Costs often not known
- Costs usually seen as significant; few were able/willing to cite costs
- Split between near term (4) and longer term benefits (7) [Includes those who cite both]

Exhibit II-14

Success/Failures

- Question which respondents had hardest time answering
- Prior failures sometimes led to more attention to system building process (3)
- Many believe that current process contributes to success
- Failure may sometimes be due to "expectation management"

Exhibit II-15

Other Organizations To Watch

- Andersen (5)
- Microsoft (3)
- TI (3)
- James Martin (2)
- KnowledgeWare (2)
- Others
  - Borland
  - IBM
  - DEC
  - "Big 6" (other than Andersen)
  - Lotus
  - Cambridge
  - Next
  - USC - Software Engineering Institute
  - Users (5)

Exhibit II-16

Charging for Higher Value Software Assets

- May not be a higher value asset (3)
- Not sure how to convince customers (5)
- Don't know how to compute value (3)
- Are addressing issue (1)

CHAPTER III:

Summary Matrixes

1. CHANGES IN THE SOFTWARE BUILDING PROCESS (OVERALL)

Borland

Techniques: Object libraries
Development Time: Reduced
Cost Reduction: Reduced
Quality: Increased

Cambridge Technology

Techniques: Rapid Prototyping
Development Time: Speeded up 3 times
Cost Reduction: Reduced
Quality: Increased

Cap Gemini

Techniques: JAD, RAD, partner tools
Development Time: Reduced by 25%
Cost Reduction: Somewhat
Quality: Some increase

DACOM

Techniques: Process Modeling
Development Time: Probably decreased
Cost Reduction: Probably decreased
Quality: Much higher

DEC

Techniques: JAD, RAD
Development Time: Reduce by 33% (goal)
Cost Reduction: Somewhat
Quality: Some increase

EDS

Techniques: JAD, RAD, Time Boxing, Function Points
Development Time: 30% less
Cost Reduction: Significant
Quality: Greatly increased

HP

Techniques: Focus on client/server
Development Time: Reduce by 25%
Cost Reduction: Reduced by 20%
Quality: High priority

IBM

Techniques: Many
Development Time: Reduce by 30% (?)
Cost Reduction: Reduce by 25-30%
Quality: Important

Microsoft

Techniques: Object libraries, JAD
Development Time: 75% less
Cost Reduction: 50% in maintenance (forecast)
Quality: Increased

Oracle

Techniques: Rapid prototyping
Development Time: Reduce by 30%
Cost Reduction: No
Quality: Zero defects (goal)

Xerox

Techniques: JAD, Objects
Development Time: 20% less
Cost Reduction: None cited
Quality: 30-50% fewer bugs

2. SOFT BENEFITS

Borland	•	Up front planning
	•	Better maintenance
Cap Gemini	•	Help understand how business processes will change
DACOM	•	Benefits accrue to underlying business
	•	Reduced cycle time, improved maintenance
	•	End users are part of team
DEC	•	Better communications
	•	Improved customer satisfaction
	•	Increased responsiveness to customer needs
EDS	•	Transform business process
HP	•	Improved customer satisfaction
	•	Reduced cycle time for business transactions

IBM	•	Improved communications
	•	Improved customer satisfaction
	•	Customer review of systems
Microsoft	•	Business Re-engineering
	•	Selling true Solutions
	•	Improved satisfaction
Oracle	•	Improved communications
	•	Replacement of paper with electronic interactions
Xerox	•	More discipline in applications building process
	•	Improved customer satisfaction

3. RELATIVE CONTRIBUTION OF TOOLS AND PROCESSES

Borland	**Tools:** Object libraries, object development and modification, GUI **Processes:**Process is most important
Cambridge	**Tools:** Important to the extent they are part of the process **Processes:** Rapid prototyping very important; parallel teams with a factory-like flavor are also important
Cap Gemini	**Tools:** Tools supplied by partners play an important secondary role **Processes:** Process is most important; some consideration to factory approach; some dependence on skilled consultants
DACOM	**Tools:** Tools are secondary **Processes:** Modeling process is key; significant dependence on skilled consultants
DEC	**Tools:** Moving away from priority placed on tools **Processes:** JAD and RAD expected to have more impact; software factories being investigated
EDS	**Tools:** IEF and INCASE (EDS); metrics **Processes:** Process is key
HP	**Tools:** CASE and other tools secondary **Processes:** Focussing most attention on process

IBM	**Tools:** Direction unclear, but tools are important **Processes:** Direction unclear
Microsoft	**Tools:** Tools are more important but still evolving; traditional tools are dead **Processes:** Teams are most important part of process
Oracle	**Tools:** Important, symbiotic with process **Processes:** Important, symbiotic with tools
Xerox	**Tools:** Higher level languages-secondary **Processes:** Organization and process are key

4. REUSABILITY

Borland	•	A focus, flowing out of object oriented
Cambridge	•	Not a focus
	•	In principle, their process of separating functions in implementation supports reusability
Cap Gemini	•	Exploring feasibility
DACOM	•	Do not see theories for supporting reusability in commercial market
	•	Have seen little potential for reusability in processes which appear similar on the surface
DEC	•	Experimenting both internally and with their VARs
	•	Have tools to support reusability
EDS	•	A long way to go
	•	Focus is on data models
HP	•	Very little being done in custom code
	•	More attention in product area

IBM	•	Experimenting
	•	Being used in system software development internally
Microsoft	•	A major expectation for future
	•	50% reusability in 3 years
Oracle	•	Reusability is an objective
	•	More opportunities in an "all Oracle" environment
	•	Need to develop standards
Xerox	•	Reusability will grow in applications development
	•	Already successful in platform migration

5. CLIENT/SERVER MODEL

Borland	**Problems Seen:** Incomplete model **Vendor Direction:** Internal efforts are addressing this
Cambridge Technology	**Problems Seen:** Incomplete model **Vendor Direction:** Integrate with reengineering
Cap Gemini	**Problems Seen:** Incomplete model [INPUT note: CGA may not fully understand issue.] **Vendor Direction:** Part of systems definition
DACOM	**Problems Seen:** Very incomplete model **Vendor Direction:** Understanding business process is critical to avoid painful mistakes
DEC	**Problems Seen:** Incomplete model **Vendor Direction:** Address technical issues
EDS	**Problems Seen:** Incomplete model **Vendor Direction:** They are evolving tools to assist
HP	**Problems Seen:** Comfortable with overall model; still need to link redesigned processes to client/server model **Vendor Direction:** Continue improvements

IBM	**Problems Seen:** Incomplete model **Vendor Direction:** Addressing separating of processes and data
Microsoft	**Problems Seen:** Model incompleteness not critical-many models are emerging
Technology	**Vendor Direction:** Education in selecting right model
Oracle	**Problems Seen:** Incomplete model, especially in heterogeneous platforms **Vendor Direction:** See standards as assistance
Xerox	**Problems Seen:** Incomplete model **Vendor Direction:** Model problems aren't standing in the way of progress

6. SYSTEM BUILDING MODELS/APPROACHES

Borland	• Single process model • Medium-high degree of management control
Cambridge Technology	• Use a single process model • High degree of management control
Cap Gemini	• Primary and secondary models • Medium-high degree of management control
DACOM	• Use a single process model • High degree of management control
DEC	• Multiple process models; efforts are coordinated • Medium degree of management control
EDS	• Multiple process model • Medium/high degree of management control
HP	• Multiple process models; coordination is attempted • Medium-low degree of management control
IBM	• Multiple process models, more being added • Low degree of management control
Microsoft	• Multiple process model now; ultimately there will be one. • High degree of management control
Oracle	• Multiple process models • Low degree of management control

Xerox	• Single process model-assessing other approaches • Medium-high degree of management control

7. SKILL NEEDS AND DISTRIBUTION

Borland	• Skills are in short supply • Try to recruit elite technical staff • May become too dependent on "people" not "process
Cambridge Technology	• Skills in short supply, distributed unevenly • Continuing education
Cap Gemini	• Skill bottlenecks • Use outside staff • Continuing education
DACOM	• Lack of skills is biggest problem • Forced to hire senior staff • Internal training is largely on the job
DEC	• Need new skills, not always available internally • Continuing education • Use outside staffing sources
EDS	• Enormous changes in skills needed • Now retraining in technical skills • Want to train in process and concepts
HP	• Technical skills are not a problem, given culture • Project management biggest gap

IBM	•	Skills are in short supply; cutbacks have made situation more acute
	•	Continuing training
	•	Use outside contractors
Microsoft	•	Severe shortage among customers (Microsoft dependent on customer teams)
	•	Believe market forces will solve shortages
Oracle	•	Skills not perceived as a problem due to culture
Xerox	•	Skill shortage-distribution changing
	•	Technical skills more easily addressable
	•	Experience in customer-related skills biggest bottlenneck.

8. SYSTEM TESTING PROCESS

Borland	•	Becoming more modular
	•	Specialist emerging (analogous to packaging and marketing specialists)
Cambridge Technology	•	Tested at prototype stage
	•	Heavy user involvement
Cap Gemini	•	Tested at early stage
	•	Users involved
DACOM	•	Testing done at model level
DEC	•	Want to incorporate testing as part of system building process
EDS	•	A lower priority
	•	I-CASE helps (but less helpful in C/S)
	•	Users involved
HP	•	Significant changes due to client/server
	•	Testing automated

IBM	•	Changes made in tools and systems review
	•	Users increasingly brought into process
Microsoft	•	Simultaneous, parallel testing
	•	Testing time and cost will be reduced
Oracle	•	Now an on-going process involving customer
	•	Testing becoming much more complex; still involves considerable manual testing
Xerox	•	Not much change seen-still an art
	•	More investment will be placed on testing specifications

9. INNOVATIONS (SELF-ASSESSED)

Borland	•	Object-oriented focus
Cambridge Technology	•	Methodology
Cap Gemini	•	Combination of their own experience and tools from other vendors
DACOM	•	Emphasis on modeling of business processes
	•	Making customer part of team
DEC	•	Factory-like code development
EDS	•	Own version of RAD
	•	Inference-based code generation
	•	Electronic distribution/education
HP	•	Don't see themselves as innovative per se

IBM	•	Various technical initiatives, largely with partners
Microsoft	•	Reusability
Oracle	•	Methodologies for continuous refinement of systems
Xerox	•	Applying intelligence to C/S development
	•	PARC/XSOFT relationship

10. COST OF MAKING IMPROVEMENTS IN SYSTEM BUILDING

Borland	•	Long term payoff
Cambridge Technology	•	Sees payoff in near term
	•	Controls in place to balance cost versus benefits
Cap Gemini	•	Seeks to control and limit these costs
	•	This is one of the reasons for using outside partners
DACOM	•	Very high costs, especially relative to their size
	•	Long term payoff
DEC	•	Long term benefit expected; short term benefit hoped for
EDS	•	High investment costs
	•	Long term payoff
HP	•	Pulling together the client/server process model cost up to $1 million in addition to uncounted staff time
	•	Benefits are immediate once the learning curve is out of the way

IBM	•	Cost is unknown
	•	Near term and longer term benefits expected
Microsoft	•	Significant costs
	•	Long term and short term payoffs
Oracle	•	Cost is unknown
	•	Viewed as long term investments
Xerox	•	Major investments required
	•	Unclear time to payoff

11. SUCCESS/FAILURES

Borland	•	Delays in getting products to market
	•	Believe higher quality will pay off in longer run
Cambridge Technology	•	They see their process as responsible for their success.
	•	Do not admit to any failures
Cap Gemini	•	The believe their approach minimizes the risk of failure
DACOM	•	They see their recent success as a company due to their approach
	•	Significant failures in past - became a learning experience
DEC	•	Would not discuss success or failure
EDS	•	Many implementations done much faster
	•	Expectation management biggest problem
HP	•	Client/Server model leads to success
	•	Early implementation failures

IBM	•	Success has been mixed with failure; due to the fragmented nature of these efforts lessons or patterns are hard to draw
Microsoft	•	Failure have been publicized
	•	Expectation management a problem
Oracle	•	Out of Oracle's public failure in introducing applications products came the changes that are now producing success.
Xerox	•	Have been able to move products from proprietary to open platform in two years

12. OTHER ORGANIZATIONS TO WATCH

Borland	•	Microsoft
Cambridge Technology	•	TI
	•	Andersen
Cap Gemini	•	Big 6
	•	Andersen
DACOM	•	James Martin Associates
DEC	•	James Martin Associates
	•	KnowledgeWare
	•	Andersen
	•	Cambridge Technology
EDS	•	TI
	•	USC-Software Engineering Institute
	•	Andersen (unsure)
	•	IBM (unsure)
HP	•	Dupont
	•	Farmers Insurance
	•	Microsoft
	•	Andersen

IBM	•	Microsoft
	•	Next
	•	TI
	•	KnowledgeWare
Microsoft	•	Lotus (in past)
	•	Borland
	•	DEC
Oracle	•	Several corporations (confidential)
Xerox	•	None cited

13. CHARGING FOR HIGHER VALUE SOFTWARE ASSETS

Borland	•	Currently addressing: Modules; pricing adjusted for different markets
Cambridge Technology	•	Believes it would be difficult to charge a premium
Cap Gemini	•	A premium might be possible in a long term relationship
DACOM	•	Do not think that customers will be educated enough to pay a premium for many years
DEC	•	Not sure how premium could be computed
	•	Would be necessary to guarantee the higher value in some way
EDS	•	Benefits = enterprise wealth/value added
	•	Not sure how to charge for benefit
HP	•	Doesn't believe that higher value assets will in fact be created
	•	New kinds of assets may be created

IBM	•	Not sure how additional value can be made clear to the customer
Microsoft	•	Values will increase as systems are tied closer to business functions
	•	Don't know how to measure
Oracle	•	Believes asset value of software will decline because of shorter life
	•	"Information assets" may increase but unclear how to measure value
Xerox	•	Will have to be measured in terms of business contribution
	•	Same software could have many different values

CHAPTER IV:

Summary By Category

Changes in the Software Building Process (Overall)

Borland

There are three reasons why Borland is changing from procedural to object oriented development.

- The software that results is more reliable; it takes less time to debug and fix code.
- The code that results is more suited to being reused.
- It is easier to make revisions and enhancements in the future.

In effect, the time and cost required to modify software can be reduced. Other benefits such as the reduction of elapsed time for a project or cost of developing new software has not or can not be quantified.

Cambridge Technology

They have introduced changes to bring about more rapid development and will add to or refine them in the future. They have speeded up the development process or time to develop an important or strategic application system by a factor of 3 (time to completion and use of the system) compared to similar projects at a client or other company where traditional methodologies or even some newer methods were used. (This appears to be so, but is based on using people with considerable knowledge and experience employed who impose a methodology that forces higher level user executives to provide timely and meaningful contributions to the process.)

The process allows the cost of the software building process to be reduced (including the person months involved) and the quality to be increased.

The development technique includes a well planned and executed preliminary requirements definition, rapid systems prototyping methodology to pin down requirements and parallel development teams to implement the system. (Top user executives are forced to provide timely and dedicated reviews of the results of the prototype through the activities of CT.)

Cap Gemini

They have ongoing programs concerned with improving the software building process, although their main concern is satisfying the needs of clients in specifying, delivering and installing business systems. They focus attention about improvements in the software building process on activities of software engineering firms such as Intersolv and Bachman who have been business partners.

Changes in the process which would speed up the development of business solutions, increase quality and/or reduce costs could increase competitiveness and be important to us. Improvements are being obtained on a regular basis, but they would like to see a substantial or notable change that could reduce the time for large, complex projects by 25% or more. This would gain attention from clients and prospects. However, control of change to their process is most important.

DACOM

The current approaches, especially CASE, haven't just reached a dead end but have lead us down the wrong dead end: The problem isn't producing code efficiently and correctly, but designing the right systems in the first place.

DACOM began as a CASE-oriented, essentially product, company in the early 1980s. It built reasonably good tools and was (and is) a leader in repository thinking. However, over time DACOM saw that the chief need was for an understanding of business processes and making improvements in the systems which support business processes.

DACOM has evolved from a company with an emphasis on software building to one that works closely with clients to design the right system in the first place. Software "implementation" is viewed by DACOM as almost easy once the underlying business design is set.

Overall time to complete may or may not be shorter compared to conventional methods. Much depends on how well their clients have been educated in the DACOM method. Initial projects are usually slower because of education and cultural change. Subsequent projects are probably shorter; however, this is very difficult to measure since historical measures of time to complete don't do justice to the time spent on the up-front business engineering phase.

Actual software development is much faster after DACOM's work:

- There are relatively few downstream changes
- The "specs" are unambiguous

Quality is immeasurably higher, although, again, there are few benchmarks and measuring methodologies for quality. This is very frustrating for DACOM because they and their clients strongly believe this to be true, but it is very difficult to prove. Other vendors make similar claims -- which DACOM doubts are true -- and DACOM's real achievements (to them) get lost in the general noise level.

DACOM admits that they could do a much better job at quantifying benefits, in order to:

- Measure their own progress
- Assess the relative strengths of different approaches
- Market themselves better

Right now, too much of their acceptance is based on clients buying into the DACOM "religion". This is ironic and frustrating for an organization that views itself as a rigorous, engineering-oriented company.

Digital Equipment

Changes are being considered or made to speed up the development of software as well as to reduce its cost and improve quality. Many separate activities are taking place ranging from steps in consulting and systems planning to work at all levels with tools and methodologies. Specific goals have not been quantified or can't be discussed, but it seems that the speedup being sought at this time would reduce development time by one third on larger projects.

EDS

They see very few changes in the "old paradigm" i.e. maintenance and minor improvements to existing legacy systems. However, they believe that for new development the changes are dramatic, both in methods and technology. The shift is from monolithic systems designed in the IBM model to C/S and open.

- "In the IBM model we forced the system to meet the capabilities of the technology."
- "In the new model we mold the technology to the system requirements."

The new approach takes an enterprise view with much more engineering orientation. The old was more artistic. They are firm believers in the 5 level Carnegie model of software development capability maturity: (1) Ad Hoc, (2) .., (3).., (4).., (5) Optimizing phase where the manager doesn't worry about the quality of the code, but rather adherence to the process. EDS feels that most commercial organizations are operating at level 1 or 2 on the scale.

Three other significant developments which are changing the way software is built are:

- Away from the total "top down" waterfall approach, from high level design to detailed implementation to "spiral" approaches which iterate to a solution. In essence, the RAD (Rapid Application Development) approach.
- The use of JAD (Joint Application Development) with users and developers sharing responsibilities throughout the development process.
- "Time Boxing" - an approach where a fixed amount of time is allowed for specification development. At the end of that period, implementation is done putting usable functionality into place. The process is then repeated on the existing system to obtain higher levels of functional refinement

Benefits Include:

- Productivity

  EDS measures productivity through the use of function points. They see this formerly proprietary approach maturing into a set of industry standards through the International Function Point User Group (IFPUG). EDS is now bidding some jobs on the basis of function points. Utilizing the new methodologies coupled with I-CASE where appropriate, they claim that EDS can get significant improvements in terms of function points implemented per hour compared to traditional methods.

- Shorter Time to Complete

  Can complete discrete applications (at the departmental level) in 70% of the time that it use to take to get to operational status. The reduction in time to complete is less for full applications suites, or enterprise wide systems. The methodology described in the attachment will begin to attack that problem.

- Quality

  Faultless code - increased functionality. EDS has implemented a standard metrics system (architecture for capturing metrics) for getting at the quality measurement problem, but it hasn't been in place long enough so that they can make any definitive statements regarding quality comparisons based on methodology deployed.

- Reusability

  Where CASE tools are applicable to the development being undertaken, a fair amount of reusability can be anticipated. However, most CASE tools are far behind in dealing with interface development, objects, and multimedia, all of which are becoming an increasing part of systems requirements.

  They are trying to develop a corporate data base on quantification of benefits, but the number of efforts which have been submitted to the metrics data base is small. When they get this into full production, they anticipate that there will be different levels of benefits - from cost saving to enterprise-wide benefits. EDS has shifted its thinking from its former view (primarily as an outsourcing vendor) of optimizing operations, to optimizing the development process. This is where they feel the real benefits will come.

Hewlett-Packard

The driving force behind changes in the systems building process at HP is to reduce the time required to deliver projects while improving the overall quality of the finished product. HP's area of emphasis is on the management process for building systems, not on tools. They are placing emphasis on the re-engineering aspects of the business process, and simultaneously trying to develop a methodology which will apply a rigorous discipline to the resulting systems implementation.

They do believe that tools are important. However, they believe that 90% of the errors occurred at the front end, where few tools are of much use without an overall re-engineering of the business process.

They believe that their emphasis on re-engineering focusing on downsizing, organizational changes, process re-design within reasonable cost containment guidelines of the client, is the primary thrust of their rethinking of the systems development process.

Reduced cycle time on the completion of business systems re-engineering project, including the generation of finished applications systems - 25%. They would not give a specific example.

HP is trying to define quality as "fit to business need" within the client's anticipated time frame. Their position is that counting bugs in the code is secondary.

HP has not made much progress on reusability. Reusability per se is not a goal. Studies that they've conducted indicate that there is huge waste in the current process which could be significantly reduced if a set of standards could be developed along with a process to better manage code; identify and select modules to be developed and maintained as "reusable", etc. However, they feel that the process is a long way off.

The main problem is that in order to generate reusable modules, the development process must be integrated or contain components and standards which insure reusability. They think that most firms are farther behind in this area than they are in CASE. (In HP's view, most firms aren't very far along in the effective integration of CASE into the development process.)

Very little is being done to obtain any more meaningful quantification of benefits than was done in the past. This process becomes even more complex when significant business re-engineering is involved. In some instances various components could be readily quantified, but that inmost instances where business investment on the basis of much softer criteria.

IBM Corporation

There are changes being made by groups in a number of areas including professional services, external consulting, SI projects, ISSC and internal software groups. IBM's interests include business systems planning, project management, methodologies and tools. Separate activities are also being carried out with business partners.

Internal professional services and software groups are under pressure from users to speed up the development process or time to develop an application system. The reduction of development time by 50 to 100% is mentioned by users as an objective although any meaningful reduction in the overall process (e.g., 20 - 33%) would be well received. IBM recognizes these objectives, but does not have a specific target. However, IBM recognizes that changes of this magnitude will require consideration of all aspects of the process from concepts of business analysis and re-engineering to project management and code generation.

Users also want to reduce the cost of developing software for a system. The reductions being sought by users are at least 30 to 40% and as much as 100% Although there is not any official target, IBM research in Endicott, NY has talked about helping users achieve a reduction of 25 to 30%.

Another objective is to allow the time and cost required to modify software to be reduced.

A different type of objective is to develop software with a higher level of quality. In addition to programs that can be more easily modified and understood, two concepts were mentioned in regard to this. Users have stated that they are interested in systems building that would automatically incorporate various types of input data validation and other cross footing that is suitable for the type of application being developed. The second concept involves enriching the development process or adding tools which will check out the logic of software during the process and leave a structure in place for acceptance testing and testing of modifications at a later date. Auditing should also be facilitated.

Many ideas about quality are prevalent. One recommendation mentioned by users is to ensure that developed software should not be fragile or become fragile when normal types of changes are incorporated or added. Users also feel that procedures or tools should guaranty that software modules can be reused or used as an object(s).

Microsoft

The major change is the movement toward larger objects. This enhances reusability. They have customers who will treat the "entire" of Excel as an object in an application. Objects were considered building blocks a long time ago, but now the ability to embed whole applications environments into a system is what is happening now.

The second biggest change is that within two years there will be a new channel in the market place that will cause market shifts -- selling "solution based objects". At first this will only happen within a given software architecture ... Microsoft, Oracle, etc. However, as standards continue to narrow down (e.g., DDE -- Dynamic Data Exchange), these solution objects will be architecturally open; the openness will be limited to a small number of popular architectures. Entire shrink-wrapped applications will be capable of being embedded into solution packages.

The third biggest change is that there will be new methodologies to facilitate the use of these objects.

The benefits fall into the following categories:

Productivity/Shorter Time

Microsoft believes that the main criteria is the time to complete development in the case of information systems and the time to market for software object and package developers. Microsoft estimates that the applications built through some sort of object methodology with reasonable tools can be completed four times faster than through the use of traditional methods.

Quality

Fully tested larger objects means that there is less chance for code test failures.

Graphical interfaces means high levels of customer satisfaction. Embedded help and AI techniques will ultimately reduce the cost of support while making an order of magnitude improvement in quality.

Reusability

They don't know exactly, but believe that it will be orders of magnitude increase shortly. They have been working with a specific customer where credit approval is used in several applications. The original module was built in ACCESS. Microsoft says that they are able to drag and drop the credit module from one application to another and through DDE link the appropriate variables between the object and the new application.

Their outside estimate is that there should be 50% reusability within the more common cross-industry functions within a few (three) years. This high level "solution" object concept coupled with better object linking and C/S design tools could generate even higher levels of reusability.

Quantification of Benefits

Microsoft sees their corporate clients increasingly quantifying benefits, but each in their own way. Microsoft is taking part in the metrics standards group. In spite of this involvement, Microsoft still believes that time to completion will still be the most important measurement, coupled in some way with a measurement of the ease of modification and maintenance.

Microsoft has done some research on 25,000 applications that have been done with their products and they say that in at least 50% of these cases a quantification process was used as part of the justification.

Oracle

The principal changes taking place in the process are driven by three fundamental driving forces.

- A migration from IS to the end-user as the leading player in the development process.
- The maturation of a new life cycle based on rapid prototyping, utilizing increasingly sophisticated tool sets.
- A growing demand for rapid turnaround and responsiveness for changes to whatever system is in place.

Both downsizing from the mainframe and upsizing from standalone departmental systems to departmental work group systems is occurring simultaneously. In either case, the traditional life cycle that was used for mainframe development is or will be replaced by the cycle described below.

- Rapid prototyping of a systems basic functionality. (Joint effort of specialists - either vendors or IS - and end-user experts.) The prototype then goes into limited production.
- Refinement and rollout of the full blown system. (Joint effort of specialists and end-user experts, and based on feedback from operating the prototype in production mode.)
- Integration of the refined system with related systems. (Primarily a specialist implementation activity based on requirements identified while operating the refined system in production.)

In other words the traditional concept of a formal requirements phase is replaced by three less formal requirements definition phases. In fact each phase of the traditional life cycle has a three counterparts in the new cycle; design, development, testing, etc.

Oracle sees itself as being involved in this process in several ways:

- As an innovative tool supplier
- As a teacher/coach
- As a high level consultant and/or implementor
- As a developer of applications software (for internal use or, perhaps, in partnership with a client)
- As a supplier of flexible applications (where Oracle may again become involved as a teacher, consultant or implementor)

These are complex roles and Oracle is the first to admit that everything isn't sorted out yet.

Meeting the real business requirements of the end-user, and shortening the time from inception to delivery in order to capture user benefits early. This fundamentally comes down to:

- Reduced training time
- Shortened development cycle to first usable version of the system
- User self-sufficiency in terms of on-going modifications, etc.

Oracle does not believe that there is likely to be a reduction in systems development costs (at least in terms of reduced manpower) over time. The more likely scenario is that implementations will have four to five times the functional capability for the same cost.

Oracle believes that the creation of new interactive applications using their redefined life cycle and tools can be accomplished in 70% of the time it would take to achieve the same functionality using conventional CASE and methods. For batch based applications or the generation of new reports, etc., they believe the number is 25%. (See attached diagram showing how newer approaches to system building generate benefits.)

Quality is customer satisfaction with the provided functionality, and "0 defect code".

In a homogeneous environment (Oracle Platform) they estimate that a fair amount of the code can be re-used. But this is not an objective unto itself. In heterogeneous environments (say using DB2 along with Oracle), less will be achieved in terms of reusability.

Oracle believes that the industry is a long way off from obtaining reusability on any major scale through the development of "objects". Probably the highest level of reusability comes through the creation of user "customizable" applications software products.

Very little quantification of benefits has been done. Most end-user projects (which are an increasing proportion of the total) seldom undergo a formal justification or subsequent audit process.

Texas Instruments

The major activity at TI is a total restructuring of the development process across all organizations. The corporate organization is managing this change in process. They've have been working on this for about one year. They are building their process on the Software Engineering Institute (Carnegie) model. Three key components to the model that they're emphasizing; (1) Dynamic (living) specification systems, (2) New quality assurance processes, (3) Configuration control and management.

From a corporate viewpoint they're interested in controlling an overall template for the development process. Each division will be allowed (within guidelines) to modify the process according to their specific needs.

- What benefits:

  - **Productivity**
    They have no history against which to measure productivity improvement. They are currently populating a data base which is capturing their results with the new approach. However, TI believes from the top down that a consistent adoption of the new methodology will yield consider savings. (The new methodology contains more sophisticated metrics which focus on user requirements in addition to productivity from the viewpoint of system development.)

  - **Shorter time**
    Have experienced decreases over originally anticipated schedules. Less overruns, but can't document improvements based on historical experience with comparable systems.

  - **Quality**
    From the IS viewpoint, there is considerable improvement. They've installed some systems which went in "bug-less", and seem to require considerably less maintenance than most. Although TI hasn't figured out how to measure it, they believe that there will be a two tiered measurement system: Technical quality and compliance with user needs and expectations.

  - **Reusability (how much?)**
    Have not had time to focus on this area. It's a longer term objective, but have done practically nothing to formalize an approach to reusability. They feel it is pointless to address this issue until they had gotten a common process in place. Furthermore, most of the work that they are involved in internally is strictly custom. There are probably not very many applications modules that are reusable.

- Quantification of benefits?

  In terms of the cost of implementing the new systems building process, a superficial cost justification was done base on other firms' data who have adopted the SEI approach. However, top management is willing to foot the bill, just because it makes sense. (Their initial successes have provided additional reinforcement to this concept.) She also said that from the point of view of internally developed applications, line management is pretty serious about justification from the top to the bottom.

Xerox

The principal changes are taking place in the process and are driven by three fundamental driving forces.

- The use of higher level languages in the creation of products (C++)
- The switch to object orientation (even if on a proprietary basis)
- Innovations in the systems/products building process itself.

Changes in the process include:

- More structured disciplines on the front end. Less "adhocticity" (their word)
- Heavy emphasis on joint design (user/designer)
- Heavy emphasis on quality
- They estimates 60% of their efforts are spent in the design phase (to prototype), 40% on actual implementation of code.

They have reduced time required by 20% to migrate existing products to new platforms: proprietary RISC to open RISC to Intel, etc. The first roll over to a new architecture took 1.5 years, the second, about 1 year, and the most recent just 9 months.

Quality improvements are along several dimensions,

- Reduced defects - 30-50% less bugs using newer approaches and tools.
- Reduced severity of defects - They use a scale for calibrating the level of defects from the design level to code errors. They wouldn't quote a number for this measurement but implied that they actually find very few code errors or design errors any more (the high and the low ends of the scale). The most severe errors they were turning up were due to anomalies in the operating systems for various platforms they support for their products.
- Improved customer satisfaction. They make extensive use of customer satisfaction as a measure of quality. They survey every six months and re-survey customers as their use and maturity level with the use of the products grows. They claim that customer satisfaction has gone up, since they have adopted new development approaches. They didn't (or couldn't) cite comparative statistics.

The extent of reusability varies.

- Looking at it from one angle, they get very high levels of reusability on platform migration efforts. Essentially all the core code moves over without modification. The primary areas where they need to custom design and develop come at developing the user interface management and integration at the operating system level.

- In terms of traditional projects where a new system/product is being developed from scratch, they are beginning to reap some of the benefits from their investment in object development. However, most objects are still at a "low" level; i.e. not very complex. XSOFT anticipates that as the number and richness of the objects grow, they will be able to obtain significant levels of reusability. Also this process will be non-linear..."richer objects will spawn even richer ones" (quote).

They don't believe that users do much in the way of quantification of benefits. Generally, PARC's offerings are not specific with regard to applications functions. They believe that when a firm decides to adopt the use of XSOFT products, it is a strategic decision, and most likely justified by functionality required to meet some business critical application decision.

Soft Benefits

Borland

Object orientation forces developers to do more up front planning in order to avoid excessive complexity.

Object orientation provides a more powerful type of development in which developers can work with a reasonable number of blocks rather than with millions of lines of coding - which has to reduce the complexity and time of development.

Cambridge Technology

The use of rapid prototyping during the requirements definition helps to improve communication about requirements as well as to analyze what is going on in business processes. If business systems are being re-engineered, concepts can be tested out. The prototype software enables users to review systems and gives them a feeling of satisfaction with the final implementation.

Cap Gemini

Improvements in software building should include steps to aid users to review how business processes will be changed or re-engineered as well as to review data management and utilization. Improvements in software engineering tools have aided both of these objectives.

DACOM

DACOM would take issue with the concept of "softer" benefits. The benefits are softer only from the standpoint that many (maybe most) of the benefits do not accrue to software building, but are gained by the underlying business. (DACOM agrees that it would be very powerful to measure the "before" and "after" business costs and benefits, but it does not do so on a consistent basis. Its clients don't want to pay for this kind of activity, because doing so won't help solve their particular business problem.}

DACOM works largely with true end users in the manufacturing/logistics area. From the standpoint of DACOM's clients the current system building process is "intolerable" and not just for the usual reasons of lateness and expense. The more thoughtful clients (and DACOM appeals to thoughtful clients, for better or worse) are frustrated by software that

- Doesn't meet their business functional needs very well.
- Behaves in inconsistent and unpredictable ways; users are overwhelmed by system details; computer systems make it harder rather than easier to understand underlying problems.
- Is difficult to change in the terms of time and expense and, even then, often can't really be changed to meet user needs

Downsizing of organizations and of the computing environment multiplies these problems

Their clients don't really care about the usual measures used to measure a "good" software project.

Digital Equipment

There is a definite need to achieve other, softer benefits for customers including improved satisfaction with the development process, improved communications on the work being done, and more ability to respond to needs. These needs are being investigated in relation to systems integration and professional services work as well as in relation to internal analyses of systems building.

EDS

> "The hard stuff is easy, the software stuff is hard. Soft stuff is harder to get than the hard stuff." Quote aside, they feel that the biggest benefit on the soft side is the benefit obtained through the change management aspects of new systems implementation methodologies: i.e. if done properly major new implementations transform the business process, either directly or indirectly.

Hewlett-Packard

HP's experience has been that the two dominant soft benefits that firms are using to justify re-systemization (and business re-engineering) efforts are:

- Improved customer satisfaction; and
- Reduced cycle time on core business transactions

Certain of HP's customers have taken to trying to measure both of these factors; but it's still difficult to relate the measures to the bottom line performance of the firm.

IBM Corporation

Software should facilitate efforts to communicate as well as to analyze what is going on in business processes. There should be a view(s) of software that enable users to review systems and give them a feeling of satisfaction with the software.

Microsoft

Microsoft cited a situation where a client of theirs was able to double the number of transactions processed per resource with a 100% improvement in measured customer satisfaction. They also cited a telecommunications company which believes that it is saving $50,000/day through a re-engineering of the process for allocating "transient" (movable) cells for peak calling periods using C/S technology. To Microsoft, these are examples of how this type of technology can improve communications within an organization.

Oracle

As firms begin to leverage the concept of information distribution and collection utilizing the interactive capabilities of modern technology significant benefits will be realized in terms of better communications both deferred and direct. This will include interaction between individuals in a firm as well as between an individual and the corporate repository of knowledge. Applying these same systems design concepts to interactions with customers, etc. will bring another whole set of benefits.

The typical knowledge worker spends between 10-15% of his/her time today interacting with paper. As the level of integration of systems goes up and the ability to interact on an as needed basis with the corporate repository grows, knowledge workers will spend probably 20-25% of their time working with information electronically.

In Oracle's view we're just seeing the tip of the iceberg in terms of understanding, articulating and realizing some of the significant benefits that can be achieved through the migration to interactive systems at all levels.

Texas Instruments

Probably the real payoff here is in quality of the products and services that TI can deliver. Very little measure, but many claims; and recently some "user" sponsored endorsements. The biggest qualitative benefits have been achieved due to the improved communications that occurs between developer and users in the systems development process. They have examples of five projects which have been delivered under the new process where the user client expressed total satisfaction with the delivered system, and in fact, had no immediate requests for changes or refinements.

Xerox

PARC/XSOFT utilize an executive council made up of key users of their products to help them decide on new directions and additions in functionality that should be made to existing products. This organization frequently cites various kinds of soft benefits as the reasons they feel that new functionality should be added. This organization provided forceful arguments as to why the product line needed to be migrated to other platforms and open operating environments. But in the end PARC and XSOFT made the decision based on their estimation of increased market share for their offerings.

Relative Contribution of Tools and Processes

Borland

Borland has focused more on the process than on tools. The use of tools, including C++, Object Windows and libraries, is important in relation to support the process.

The process involves using a library of objects, tools to help modify old objects and develop new ones and tools for linking objects. A GUI is used with the process and it has aids to ensure good practices are followed but high level experience/skills are needed.

They can also add other tools such as expert systems that will allow developers to check off boxes as part of the process.

Cambridge Technology

Both tools and processes are used in the CT Partners methodology. The rapid prototyping capability is more of a tool-like technique that allows concepts to be tested.

The development of the implementation code by the use of parallel teams is more of a process or factory like method in which one team handles the application, a second handles connectivity, a third the user interface, and a fourth the data.

Cap Gemini

As noted above, tools have brought ongoing improvements in the software building process, but changes in the overall process have made the greatest impact on their ability to deliver timely and quality work, and we would like to see an improvement in the process that would provide a substantial improvement in time and cost requirements while also enhancing quality further.

They are reviewing the use of factory-like methods to generate code although most of the effort is devoted to working with users and difficulties could occur in trying to implement a factory process in that situation.

DACOM

DACOM has a full perspective, having started as a CASE company (before the term was invented) and now describe themselves as a "systems integration consultant."

DACOM does use tools, some of which are their own (see attachment 1).

However, they see tools as essentially supporting a model-oriented approach ("Tools are the finger painting of business engineering").

- They see almost all CASE tools as not supporting the improvement of business processes (although pieces of them can do so with the right orientation)
- A good repository is vital, but they don't know of one that really meets their needs.
- They are forced to use and link standalone tools. They hope that in the future there will be integrated tools that will support a range of modeling techniques.

A key component (and bottleneck) is the need for consultants with the necessary "wisdom" to integrate the business problem, modeling techniques and client change. This is the strength of their business, but also a key limitation.

- These traits make them valuable to customers, but expensive.
- DACOM would like to be able to leverage this knowledge in a more embedded methodology. They see the application of AI and fuzzy logic as possibilities, but are currently resource-constrained in being able to pursue these directions very aggressively. (They are especially cautious since in the past they have tried to prematurely productize processes.)

The general process that they use is shown in Attachment 2.

Digital Equipment

Tools including CASE tools and other software aids are being sought and evaluated to aid with the system building process. Tools may seem to be more important in regard to current improvements in systems building, but improvement of the entire process that is carried out in systems building offers more opportunities for improvement.

The use of techniques such as JAD or RAD as part of a process are expected to have more impact on time, cost and quality than tools can.

Research is being conducted that addresses both the use of tools and processes to improve the system building process. The use of manufacturing techniques in software production has been carried out and is also being investigated together with other topics.

EDS

- Process and organization

  Process is the key and the tools must fit the process. Organizations need to focus on measuring the process...not the people. They believe this is the biggest failure of most implementation processes. They also believe that much progress is being made in the "manufacturing" approach to systems implementation, but it goes in "fits and starts" depending upon the availability of the right tools to match the process.

- Tools

  From a CASE standpoint, EDS uses both IEF and its own product INCASE. Other tools exist to manage the overall process as described in the attachment. In general however, all these tools need to be updated significantly to deal with today's architectures and development processes.

  In terms of AI and embedded methodologies, EDS believes that a lot is already there, and expects to see more and more "intelligence" and "judgment" built into tools as the evolve.

Hewlett-Packard

HP believes that the biggest contributions are coming through improvements in the process and people management aspects of the systems building process. CASE tools and other automated approaches are important but secondary. They have adopted an approach which pairs top-flight project managers with tailored teams for specific implementations. Project managers are responsible for projects only. Team members are managed in the personnel and development sense by other managers within their home organizations.

- They have installed new processes that permit feedback between project managers and those managers in the firm who are responsible for the development and administration of individuals within the organization.

- They have installed a special program to continually improve even seasoned project managers skills.

- They are using some incentive programs for project performance and quality on an experimental basis, but these programs are immature in that they tend to reward heroic fire fighting efforts...rather than solid fire prevention on a day to day basis. Incentives (both team and individual will probably play a stronger role in the future.

IBM Corporation

CASE won't achieve all the objectives that are being discussed for software but CASE tools can aid by enabling software to be analyzed, modified and generated more easily for a number of applications. It was noted that certain applications that require high levels of transaction processing might require attention from personnel that were experts in the alternatives and tools available to meet tasks. One project mentioned in this regard involved an RS/6000 system that AIC worked on in conjunction with the research staff in Austin.

An expert on GUI has been hired by IBM in Endicott, NY to develop special tools to support and guide developers in implementing certain types of workstation systems faster and with more quality.

In regard to the use of tools, it is felt that AI and particularly expert systems, as well as other capabilities should be able to be incorporated as objects in the software development process.

It is felt that the process of systems development must address a much broader set of needs than tools can address. For instance, it must address changing work structures such as work groups and other changes in work organization or people management.

Ideas for the application of manufacturing techniques to software building are also being considered, tracked and experimented with although it is felt that it may be difficult to achieve other objectives such as improvements in communication as mentioned above together with software manufacturing.

Microsoft

Microsoft believes that tools are ahead of processes in terms of contributing to success in improvements in the systems implementation process. Even though Windows NT is just being released, there are already 70 tools which have been tested and will be released soon. Whether they are all good tools is another question. They do not see process(es) moving so quickly.

From an organizational viewpoint, they believe that multi-disciplinary teams are important. This is the only change in the process that they see has made a difference.

In terms of methodologies, they believe that top down methodologies are dead, including FOUNDATION and other approaches which were designed to constrain the design to available architectures. While this class of methodologies can be adapted to a degree to current design and implementation architectures, many of the fundamental propositions upon which they are based are no longer true.

Microsoft is developing a methodology designed for developing solutions for C/S environments. It will be full bore, i.e., it will start with business re-engineering and end up with code. They expect to sell this approach directly to user organizations and/or market it through professional service firms.

Oracle

New tools and changes in processes are probably making an equal contribution; and they continue to evolve in an iterative manner. The methodology that Oracle endorses of rapid prototyping, refinement and integration has stimulated the need for new functionality in the tool set. And, the capabilities that are subsequently generated in the tool set provide opportunities for further refinement in the design and implementation processes.

CASE tools are either obsolete or essential. If they are integrated with rapid prototyping methodology and deal with the concept of distributed architecture, they are essential. Tools that "assume" the kinds of structured methodologies of the 70's and 80's are probably obsolete because they don't integrate with current methodologies or architectures.

AI, like many other buzz words (including CASE), has gotten a bad name. However, if you look around, you'll see that expert rule based systems are imbedded in everything from spreadsheets and word processors to systems development tools. The use of these technologies to support the systems building process will continue to grow, and will result in both reduced development time and higher quality code.

The biggest single change on the people and organization side was the migration of development responsibility from IS to the end-user. Distributed IS has changed the process. The result is more teamed approaches which hopefully combine the required technological expertise with user knowledge of business process.

In terms of management techniques, people are still going to be people and managers will still vary in quality. However, the teaming of users and technical personnel in the development process will and should continue to reduce the complexities of management by improving communications. New tools and processes will continue to reduce the number of people required to accomplish an implementation (and therefore the complexity).

Oracle believes there is some fallacy in the notion of "manufacturing" code. Manufacturing implies making a large quantity of the same (or essentially the same) thing. Building applications involves using common processes to generate unique systems. The thing that they have in common is the use of consistent (but obviously different) processes... little else.

Texas Instruments

The process must be in place first. They have deliberately put manual systems into place first. Once there was experience with the process, they have been able to develop specifications for what kinds of tools that they need. They have since been adding the tools that the feel will give them maximum leverage.

They do use IEF, and they are primarily a large mainframe shop, but have been preparing for the kinds of processes and tools that they will need for the inevitable movement toward distributed systems.

Recently they have installed a tool to assist them with project scheduling, and are examining alternatives for C/S environments.

Xerox

Organizational approach and process are key. The process must maintain a strong linkage between the development team and the end-user through all phases-from conceptual functional design through test and customer support. This is equally true for software products as for traditional applications development. Changes in the process are seen as responsible for 70% of the improvements that they are getting.

From an organizational viewpoint, teams are important. XSOFT uses team reward structures in most development efforts. Teams are populated based on skill requirements, customer communication capabilities, and project management experience. Part of the reward structure is based on the product quality as measured by the customer satisfaction survey process.

Reusability

Borland

One of the objectives of Borland's object oriented process is to develop reusable objects. During the development of one software product, they develop or modify objects that would be reusable for other products.

Cambridge Technology

The methodology described above has not focused on reusability, but the use of parallel development efforts would facilitate the consideration of reusable code since functional aspects of coding are separated.

Cap Gemini

Cap Gemini and its business partners are exploring the use of reusability of objects as well as object oriented programming, but this is one of many efforts to improve methods.

DACOM

They are attracted to the concept of reusability but have found only a small number of "paradigm processes" which they can see as potential candidates. For example, "inventory" should qualify, but doesn't once the major differences between industries are examined as well as the "flavors" between firms.

They see no theory of reusability in the commercial world (unlike the formulas in the scientific world) to make their job easier. They use common skeletons and analyze differences but don't really view this as reusability.

They reuse "wisdom" and are aware of the deficiencies in doing so.

Digital Equipment

Tools are being used that allow a repeatable solution to be sold. For certain applications, these tools will allow the solution to be modified or customized more easily, and many components of the solution would not have to be changed. There have been various types of reusability in the past by DEC VARs as well as by DEC. Digital has also been experimenting with reusing objects in a group of internal projects and is now doing that on a regular basis

EDS

There are varying degrees of reusability from program elements, routines, shells and now objects. Even though EDS is making heavy investments in object technology, there's a long way to go. The primary problem is knowing what's in the object library and precisely what it does. In other words the conceptual approaches to object management have a long way to go. An EDS saying is that most object library systems are like a roach motel...a lot check in, but damn few are heard from again!

The most reusable parts of most commercial systems are likely to be the data models. They've had great success in this area. In other words, processes may vary from one firm to another, but the fundamental data elements and their relationships to each other in most operational systems are the same, or close enough, that significant productivity can be gained by migrating data models between like applications.

Hewlett-Packard

Very little is being done by HP in the reusability concept as applied to custom code development. However, a significant amount of work is going into refining process which will insure reusability of some significant portion of the code that's developed for applications or systems software products.

What is being done in this area is proprietary to HP and its products. They are still a long way from the point where the generation of "universal objects", etc. will be commonplace

IBM Corporation

Reusability was mentioned several times as an objective in relation to the goals stated above. The objective was stated as creating software that can be reused and/or facilitate the reuse of other software whenever feasible.

Ideas have been tried out internally by IBM in Boca Raton, Austin, and other locations to reuse software during systems software development.

Research projects have also been carried out to test reusability and other concepts and to track user and vendor work with reusability.

Microsoft

See point 1 above.

Oracle

From Oracle's point of view, reusability is an objective. However, a lot of effort will need to be placed into the development of standards, and the consistent use of sophisticated tools before this will be accomplished through "object" libraries. In the meantime, Oracle targets on being able to reuse higher level application definitions to "regenerate" code. Modifications to the definitions at the high end allow customization; and the use of tools allows new custom code to be generated at minimal cost.

There may not be much progress was likely to get made in generating reusable modules at the user interface, since this is an area where systems providing similar functionality were liable to vary significantly based on personal and organizational preferences.

Texas Instruments

(See Question #1)

Xerox

(See question #1)

Difficulties in Addressing Client/Server

Borland

Difficulties involved in separating processes or splitting the management of data can occur and require planning that might, for instance, involve the use of a common data base. These difficulties are addressed in internal development efforts.

Borland also has a product, Interbase, now in use in the Philadelphia stock exchange, that addresses the need for a database that can be used by multiple systems. Borland referred to Interbase as middleware.

Borland sees itself as a client/server oriented firm.

Cambridge Technology

There can be difficulties since corporate objectives as well as processes and data can be divided between uncoordinated systems on client/servers. Cambridge Technology allows client/server systems to be handled by themselves or better yet, together with other, related systems changes that are necessary to meet corporate goals. The methodology utilizes open systems which facilitates work with a variety of client/server technology.

They also test out application approaches with rapid prototyping that should highlight questions about the segmentation of functions and data across systems.

Cap Gemini

The needs of users have to be fully understood before decisions are made regarding the technology that will be used in a solution. Decisions are made too quickly in regard to the use of client/server technology, and problems can result in relation to data management and the division of functions between business units.

DACOM

The client/server model is extremely incomplete. More basically, there is no model for organizational or functional downsizing. This puts a process engineering in a double bind. Learning for DACOM is "painful" and largely intuitive as far as where data, processing and location boundaries are drawn.

This disturbs them less than may appear, since the analytic process involved in understanding the business processes will usually keep them from making unrecoverable mistakes in "client" and "server" function allocations.

They feel great sympathy with their partners (whether customer-partners or vendor-partners) who are trying to make client/server technology work.

Digital Equipment

There are questions about the allocation of data and processes from a consolidated system to a number of client/server systems. There are also questions about using a set of data or transactions received at one point to update multiple client/server systems.

EDS

Yes, the client/server model is incomplete particularly at the high level; i.e. there are very few universal models for the design phase of client/server applications; i.e. allocation of processes and data between the client and server. This is particularly true where portability across multiple architectures is a system requirement.

However, EDS believes that the CASE tools are evolving to deal with these issues on a static basis. The real long term objective however, should be to design C/S systems which can reallocate functionality and data between the client and the server dynamically, based on transaction volume, or other key systems parameters.

EDS uses SES (Scientific Engineering Software Model) which helps designers forecast performance at the design phase as a function of the allocation of data and processes and transaction loading models.

Hewlett-Packard

HP has been working at the C/S model in an open systems environment for some number of years. They are very comfortable with it at this point, and have a formally documented approach for allocating functions and data between clients and servers. (As is the case with reusability, the model is proprietary to HP and might not be applicable to other platform architectures.) The model is supposed to be rigorous and has significantly improved the quality and efficiency with which new applications can be developing using C/S.

They still needed to improve the overall process at the design level. The area that needs the greatest improvement is a consistent approach to linking redesigned business processes with functional specifications for C/S systems. They are working on this.

- They have just dropped KnowledgeWare in favor of IEF as their primary CASE vendor.
- Anticipate that IEF organization will work jointly with them on tailoring CASE tools to meet HP's needs both at the design and implementation levels.

A formal model for the C/S design process has significantly improved the quality and reduced the time required to systematically test new code. Estimates that they are saving about 20-25% in this area, and are certain that the quality of the code is higher.

IBM Corporation

Difficulties involved in separating processes or splitting the management of data have not been fully addressed. IBM thinks that these problems have often led to the use of larger capabilities, AS/400s rather than work stations, in many client/server systems.

Microsoft

Microsoft believes that there will not ever be a perfect C/S model, because C/S technology needs to be adaptable to the solution, not vice-versa. The tools in place are increasingly sophisticated -- how to use them effectively is another matter. The problem is educating users and providing design processes which deal with new paradigms of information processing, including graphical and multi-media communication interfaces. Things are moving so rapidly that the problem will soon be selecting the appropriate model for the desired solution, not whether there are enough models around.

Oracle

Oracle believes that the C/S model is incomplete. When utilizing a consistent platform on any given set of applications (the Oracle platform) they have a consistent way to model the distribution of processing and data. It's pretty straightforward. Shared functions and data go to the server, and unique functions and data go to the client. However, when working with heterogeneous platforms (in particular data bases), the ability to follow the model is frequently inhibited by differences in functionality in hardware and software.

Over time, the migration to more open systems and standard interfaces will likely increase the level of consistency that can be achieved in allocating data and processing in the C/S environment. In any event, its not a serious enough problem that it is likely to inhibit the ongoing emergence of C/S as the primary applications platform.

Texas Instruments

The demand for C/S has been growing at TI. Many implementations have already been done. However, the model is definitely incomplete. They have 12 people dedicated to developing the methodology and tool sets that would be required to place C/S development in the context of their overall project development and management scheme. They are hoping to be in front of this situation, but there is no clear cut approach which seems to be dominating.

Xerox

Yes, the client/server model is incomplete. In particular, most applications function transfer takes place between files, or blocks of code. There needs to be more intelligence between what is transferred between the two peer level environments. In other words the environment needs to be more collaborative. Also, some model for dynamic reallocation of activities between clients and servers depending on the computing environment at any given time.

The fact that the perfect model doesn't exist is not likely to inhibit the continued migration to C/S for new applications. And, once a design decision (based on whatever allocation of functions is decided) is made, the new development approaches using objects, etc. can leverage that design.

Is More Than One Process Model to Improve Systems Building?

Borland

Borland is now dedicated to the use of one process model, the object oriented model of systems building. Change is managed or coordinated on an overall basis to a great extent (4.0 on a scale of 1 to 5 where 5 = high level of management control).

Cambridge Technology

CT Partners uses only one process model, although improvements to it are considered, evaluated and implemented over time. They have a high degree of control over management of change.

Cap Gemini

Within CGA, there is an effort to use one model although experimentation is made with other models. We are willing to support projects at users that employ models different from the one we support internally. Some individual offices or personnel are attempting to change methods, but control is achieved. They have a medium-high degree of control over the management of change.

Due to uncertainties about the systems building process as well as the lack of coordination between various department in a company, multiple models may be in use at a client, and they might be involved in supporting more than one.

DACOM

They have their own approach (attachment 2). However, they feel agnostic toward particular business analysis practices and feel it is too early to back one horse. They have a high degree of control over the management of change.

Digital Equipment

DEC is using multiple process models, but most of these efforts are coordinated. In some circumstances, activities in support of accounts could use tools and techniques or new methodologies in an unplanned manner to improve systems building. Where a VAR or associated vendor is in a lead position, this is particularly true. They have a medium degree o control over the management of change.

Tools are available to help manage or control change management at DEC, but they are not always used.

EDS

Yes, a single one won't work! In fact EDS's entire development process is based on that assumption. See attachment, "EDS Methodology Architecture". The underlying assumption is that you can force an overall architecture of framework, but the framework must be able to accommodate some variations in methodology and even more in some specific processes. Within this context, change is managed/coordinated quite well. (4.0 on a scale of 1 to 5, where 5 = high level of management control.)

Hewlett-Packard

HP is definitely trying multiple approaches to improving the systems building process, but the process is not seen internally as very well coordinated. Individual initiatives appear to pop up, and get piloted. Informal communications between people involved in these efforts are the primary method of coordination. However, enough information gets shared to insure that some "best of breed" approaches are emerging.

There is a group manager whose responsibility it is to gather, digest and redistribute information on the new techniques and approaches that emerge. Overall, they have a medium to low degree of control over the management of change.

IBM Corporation

IBM has a number of process models available to improve systems building and is constantly in contact with other vendors to make arrangements to share and try out their ideas (e.g., recent agreement with HP on client/server development products). They have a low degree of control over the management of change. Many offices (but not all) are supportive of change due to this. IBM contacts in professional services assignments feel that this emphasizes that the most important element in the improvement of systems building is the experience and knowledge of industries and application systems.

IBM points out that some large corporations such as several of the top banks are trying to use more than one process model for improving systems building. This has occurred chiefly due to unplanned pressures by users for different alternatives for systems building. There may not be an adequate means of managing or coordinating change in these situations even if IS tries to coordinate or manage change in situations where more than one model is operative.

Some areas of organizations are more supportive of change, particularly users who are under pressure to achieve business goals.

Microsoft

Yes, there will be more than one approach in the short term because there are not yet clear cur winners. They are testing the waters. Ultimately they want one approach.

Change is managed or coordinated relatively tightly (4 on a scale of 1 to 5, where 5 = a high level of management control)

Oracle

Yes, Oracle is certainly trying more than one model. Two examples have already been cited; the rapid prototyping process and the creation of "customizable" applications packages. More are likely to be tried in the future.

In terms of how the process is being managed... at Oracle it's ad hoc. Overall, they have a low degree of control over the management of change.

Texas Instruments

TI's corporate strategy is to have a single general approach. TI feels that this is working. In essence they require compliance with the overall model, but are flexible as to how individual divisions tune it and select tools that meet their particular needs. The process isn't totally top down. There is an organization that functions as a "switch" for new ideas regarding innovations that might percolate up through the divisional organizations.

Xerox

PARC and XSOFT are pretty will coordinated on managing the evolution for the systems building process. They term it a "cohesive" approach. They have some "skunk works" groups trying now new concepts, but it's not a competitive ownership issues. These organizations are funded to test out new ideas. The results they achieved are evaluated and changes are made to the core processes to leverage any significant concepts developed by the "skunk works" teams. Change is managed/coordinated very well. (5.0 on a scale of 1 to 5, where 5 = high level of management control.)

Skill Needs and Distribution

Borland

Skills to support the process carried out at Borland are in short supply. Proactive steps to improve the situation include careful hiring practices with some universities as well as internal training. Programmers and QA people will need 3 to 6 months of internal training to become useful.

New recruits are sought at a handful of universities, who do things the "right way" then, the recruits are given high level training for three to six months. The result may be a creation/selection of "prima donnas".

Cambridge Technology

Higher level skills are in short supply and distributed unevenly. CT Partners provides a method that allows the need for skills to be managed and controlled more easily.

The CT Group, an affiliated organization, provides education in the methodology utilized by CT Partners which provides more opportunity for overcoming skill deficiencies.

Cap Gemini

There are currently needs for more software development skills due to the variety of tools, languages, databases, operating systems, and other systems and application software in use for workstations as well as other platforms. CGA as well as other SI and professional services firms can not count on having the skills available that prospects and clients have in use or plan to utilize. Alliances, temporary staff members and the services of specialized temporary employment services must be used.

Skills bottlenecks are being encountered. CGA will propose changes in the software products planned or specified for jobs by users. Recently, one client was persuaded to step back to an earlier version of a workstation operating system in order to use the network manager which had been selected.

In addition to proposing changes to the set of software products selected for a project, the proactive step more often taken by CGA is to work with clients and recommend a set of products that can be well supported.

Internal education as well as relations with consultants and other firms are also sought to expand the skills inventory that CGA can support.

DACOM

Finding people who understand business processes is their greatest need. It takes several years to train a new employee with little business experience (largely on the job). They can predict fairly well the people who will not ultimately make it, after a day of pre-qualification testing; however, they can only predict those who will not fail; they are much less sure of those who will do very well. They have often been forced to hire more senior people as a result; this kind of person is more expensive, pound for pound, than a more junior person. A larger problem is that it is more difficult to form a cohesive "DACOM way" from people who are more set in their ways (even if these are sound, productive ways).

DACOM would like to emulate the philosophy of an EDS or Andersen and mold new recruits to their culture.

Digital Equipment

There are a number of new skills that are needed in current projects ranging from those needed to use new systems development and relational database tools to those available to use network software products, GUI and workstation spreadsheet and database products.

The new skills are not always present at DEC. Some internal training is carried on and external training is made use of. However, a group of contract services vendors are used to meet critical situations with temporary people. VARs and associated consultants are also used to meet critical needs.

EDS

There is enormous change in this area driven primarily by three factors:

- Demands for new kinds of applications (new processes to automate) including business support, new kinds of decision support, etc.
- New technology. In the old IBM scenario, the bottom line was that you were going to use a 3270. This is totally unacceptable today. The choices of hardware architecture alone present serious challenges. EDS has calculated that using just the most popular hardware and software products that the number of combinations of platforms for a typical C/S application could theoretically approach 7.6 times 10 to the 50th power. So the trick is to isolate the development process from the platform. New interfaces and new media also present a challenge on the training and skill development front.
- New processes of systems building.

EDS is focusing on shifting its training emphasis to accommodate these new requirements in a revolutionized environment. They are placing heavy emphasis on interface design I-CASE, C++, object development techniques and 3GL languages. And they continue to expand their Interactive Distance Learning Network (IDLN). This provides a virtual classroom environment where professionals can lock in from anywhere in the world, and using interactive keypads, participate in classroom sessions. Instructors can obtain instant feedback, etc.

Right now this capability is limited pretty much to teaching technical skills, but that new courseware is being developed which will allow people to learn about processes and higher level concepts as well.

Hewlett-Packard

Technical skills don't seem to be a problem for HP. The company's culture supports investing in employees, particularly in technical education. Project management is the primary area where they see a significant shortage. And, there are no magic wands to wave that eliminate the need for the "experience" factor in project management. They are trying an "understudy" approach which pairs less experienced project managers with more experienced ones to hopefully obtain knowledge transfer.

IBM Corporation

Skills are still in short supply and distributed unevenly. Hiring restrictions and staff reductions have made it difficult to get the right people. The IBM culture still places great emphasis on training. Proactive steps to improve the situation include arrangements with vendors that can aid with or participate in improvement in software building.

Microsoft

Yes, there is a severe shortage of skills overall in the firms that Microsoft works with; Microsoft can't supply all the skills themselves, so is dependent partly on its users and customers. The former business analyst must be transformed into someone who can support re-engineering or processes and build prototypes of applications using high level tools. More technical staff will become module or "component" builders.

The problem is significant because the formal processes, standards, tools and educational methods are falling into place to make these new skill paradigms happen.

Basically, Microsoft feels that because there is a high need for transformation and evolving support tools/processes, that the market will make this adjustment in skills occur.

Oracle

Oracle, being a technology company, makes heavy investments in training, and provides numerous incentives for self-development. On-going training and development is just part of the culture. So fundamentally, Oracle doesn't feel that it's facing a any significant problem in finding or developing the skills that it needs.

In general, Oracle sees the distribution of IS skills to user organizations is a significant change, and will continue to happen throughout the decade. In terms of skills bottlenecks, Oracle believes that these are largely transient effects.

Texas Instruments

They are finding that there is a strong need for people with specific skills. Specific skills which appear to be in short supply are business engineering, quality assurance, and project management. TI feels it is able to overcome the demand for new types of technical skills through training and hiring. They find themselves with a very limited set of people who have the non-technical experience to adapt to new methodologies. These people are harder to find, and training is not nearly as effective as on the technical side.

To overcome this problem they have developed a series of "centers of expertise". They use teaming to staff projects. They used to expect their "analysts" to be able to do everything from business analysis to systems test. They are now attempting to maximize the utilization of the scarce skills through teaming.

Xerox

Yes, there's a shortage of skills; and yes, the distribution is changing. From the actual development viewpoint, PARC/XSOFT is finding that it can do more with less people, but that to achieve the quality levels they strive for, more people time is being invested into customer related activities. The technical skill problem is easier to deal with, in light of the fact that they are primarily a technology company. However, experience is a big factor in terms of developing individuals who can handle the customer end of the process. This area is the biggest bottleneck a the moment.

System Testing Process

Borland

It is evolving to become more modular.

Testing for a given type of an engine, for instance, could always be handled by the same team. This would be similar in concept to having packaging and marketing handled by specialists.

Cambridge Technology

Users are being brought in to systems testing more, particularly to review the systems concept or business re-engineering envisioned. The methodology of CT Partners is supportive of this by enabling users to formulate requirements in a structured manner and test out the system approach more quickly through the rapid generation of a prototype.

Cap Gemini

More comprehensive tests are formulated with the aid of users, and users are brought into the review of proposed systems at an earlier stage through the use of front end CASE tools and JAD or RAD types of processes.

DACOM

Testing should be done at the model level.

Digital Equipment

Efforts are being made at Digital to develop new types of testing that will be automatically incorporated into the system building process.

There are also some products being introduced jointly with large companies that will handle portions of the testing process in new ways through system review with front end CASE tools, prototyping and other means.

EDS

EDS currently has little to offer here. They believe that a lot more emphasis is being and will continue to be, placed on testing at every phase of the life cycle. The use of I-CASE has simplified the testing process and reduced the number of bugs which must be fixed significantly. The less code you write by hand, the higher the quality. On many implementations 90% of the code will come through CASE and the remaining 10% will be written by hand. (Of course, the 10% is more difficult).

The other thing that's changing in the testing process, is the heavy involvement of users in all aspects of testing.

Hewlett-Packard

Adoption of the C/S model has significantly changed (and improved) the testing process. (See the answer to question 5.) Since the modes provides for a limited set of structured communications between client and server using standard protocols, it's possible to develop test packages to automate the testing of each units functionality separately and in most instances simultaneously. This has lowered testing costs and cycle times significantly.

IBM Corporation

Users are being brought in to systems testing to a greater extent. New types of tools or systems review are being sought to improve the ability of users to test systems. Where vendors are involved in the supply of or development of software products, more accountability is being placed upon them during systems testing.

Microsoft

Integrated testing used to be the final step in a new system. Under the new C/S model, testing will occur simultaneously across the entire development process. Individual modules (objects) will be bug-proof. Code will integrate modules into solutions and will increasingly use high level meta languages. This will reduce the risk of "hard" coding failures.

The net effect is that time and costs will become significantly less over time.

Oracle

Testing is becoming an on-going continuous process. The objective is no longer just to meet a written specification, but to insure that the customer is satisfied with the cost, functional performance and time to delivery of the system. In a sense this increases the complexity of the entire testing process, and places the "pass fail" judgment directly to the user.

In terms of actually testing finished code, major tests are built in at each of the three major phases of the Oracle process (rapid prototyping, refinement, integration).

The process of regression testing of software is growing more complex. As more and more functionality is included, the possibilities that need to be tested multiply exponentially. Some automated processes have been put in place, but manual testing of individual products is still required.

Texas Instruments

Using a more iterative approach which is consistent with the tighter coupling of users and developers in the new systems building life cycle. TI feels that as far as the code testing process, they've had an advanced system for some time. Nevertheless several initiatives are going forward to insure that whatever improvements can be made in testing, are.

Essentially, they have an "early" inspection process for testing. Same as most other modular approaches.

Xerox

They don't see much change in process itself. More emphasis is being placed on testing specifications, although this is still more of an art than a science. Over time they expect that 70-80% of the investment in testing will be in this area. Clearly there's considerably more customer involvement throughout the development cycle. But, the process itself is not significantly different.

Particular Innovations

Borland

The object oriented process they use is particularly innovative.

Cambridge Technology

CT Partner's believes its methodology (described in section 1) is particularly innovative.

CT Partner's also feels that its ability to rapidly develop strategic systems using an open system approach is innovative.

Cap Gemini

The effort to use the software engineering tools and concepts of leading software engineering firms together with the experience of CGA is felt to be innovative. Some other SI/professional services firms concentrate on using their own tools and methodologies. CGA wants to incorporate the ideas of firms who specialize in improving the software building process.

Position papers and internal training ensures that innovations are spread throughout CGA. CGA is prepared to work with clients to help them speed innovative changes in systems building, but this can be difficult to achieve in large organizations, since many user groups are acting independently.

DACOM

DACOM believes that their entire approach is innovative, i.e.,

- Emphasizing modeling the upstream business process and defining objectives
- Focus on model building
- Making customer part of team, involving high levels of education and training
- De-emphasize conventional "software building" part of the equation (even though many of the DACOM staff has a heavy background in software building).

Digital Equipment

Digital feels that its system building process is innovative since it results in complex network solutions in a relatively short time.

Several recent internal efforts have resulted in factory-like development of code, but can't be discussed yet.

EDS

EDS feels its particular innovations include:

- Inference based code generation (CDM)
- Rapid interactive systems engineering (RISE)
- Electronic distribution of life cycle, courseware, and processes

EDS's approach to an overall framework as described in the attachment is also rather innovative and unique.

Hewlett-Packard

They believe that there is significant variation between major firms, but that probably all leading firms have hot spots where advances are being made. Several user companies and two vendors were cited.

- Dupont - Significant work in "rapid" prototyping for C/S applications
- Farmers' Insurance - Advanced architecture for C/S transaction management
- Microsoft - Use of advanced tools and concepts (object oriented coding, etc.) to automate the systems building process. Reusability at applications product level.
- HP - C/S applications model and transaction management
- Andersen - Re-engineering process and (probably) reusable applications modules.

IBM Corporation

Innovation of various types is being tried by corporations in contact with IBM including the development of specialized tools internally or by vendors, the generation of objects in C++ code that can be used to add necessary functions to a number of systems, the use of specialized CASE approaches, greater reliance on vendor application software products and experimentation with new types of languages. One effort that IBM is engage in with a user involves the use of specialized GUI that aid a developer to combine objects. Use of the GUI will guide (and force) the developer to include various types of controls and aids that will facilitate testing and operation of the resulting system.

IBM is also conducting research on ideas about GUI that use symbols other than icons. One of these concerns the use of symbols that would aid a person involved in a work group structure.

Microsoft

[Several case studies are promised, but have not yet been received.]

Oracle

- More and more integration of applications systems and data
- Installing methodologies which permit the continuous refinement of systems
- Providing tools and methodologies which allow users to achieve high levels of self-sufficiency in terms of managing their own systems environment.

Texas Instruments

TI considers itself very innovative in their quality assurance process. Instead of after the fact, their quality assurance process is focused on identifying failure risks up front, and clearing the way for success. The process is iterative.

Xerox

There are two areas where PARC/XSOFT sees itself as being particularly innovative.

- From a technology viewpoint - they are investing considerable effort in developing more sophisticated software to apply intelligence to the server. (This issue was discussed earlier in the interview in question number 5.)

- From a business standpoint, they believe that the relationship between PARC and XSOFT is unique and mutually beneficial. PARC places emphasis on innovative technology concepts, and XSOFT on how they can be converted into good business opportunities.

They feel there were pockets of innovation all over the industry where significant and innovative strides are being made in the way software is designed and built, but didn't think anyone had a monopoly or could be considered the industry leader.

Cost of Making Improvements in Systems Building

(Funds, People, and Opportunities Foregone)

Borland

There was a cost in terms of the delays encountered in bringing certain products to market (Quattro and dBase releases). They feel, what they accomplished will pay off in the long term since the products are more solid, less in need of fixes.

Cambridge Technology

Cambridge Technology has experience in relation to this topic. It originally sold education that concentrated on using technology to obtain competitive advantage. The cost of implementing the methods it advocated were hard to justify in many organizations. It was easier for these organizations to pay for the use of the methods to implement strategic applications more rapidly. A near term payoff on a project justified using the methods.

Since the current process is so controlled, the cost of making improvements to it can be evaluated in terms of both near and longer term benefits. No changes are foreseen that would change the process to a significant extent.

Cap Gemini

Improvements in systems building within CGA takes time and consume funds and personnel, and can delay getting work done and limit or preclude taking advantage of business opportunities. That is why CGA tries to use the experience of business partners active in systems development processes and tools as well as experiment with methods that might improve systems development. CGA wants to be able to work with clients that want to use various improvements, even if temporary consultants are employed to aid with that; however, CGA wants to limit internal changes unless there are longer term advantages. .

DACOM

High costs, both to themselves and their customers. [A strength and weakness of DACOM is the fuzzy line between themselves and their customers.] DACOM has plowed back a great deal of time and money into iterations of their approach. They believe that they are on a very productive path now. But the whole process has definitely been a long term investment for them.

Digital Equipment

It is hoped that there will be some rapid or near term improvements although most observers at DEC feel that it will be a long term investment.

EDS

The costs are high. Excluding retraining in technology and processes, the cost of properly outfitting a developer to operate effectively in the new environment is $70,000; $50,000 for hardware and software tools, $20,000 for "objects" and other development aids. EDS is a firm believer that this should be done, and believes that up until recently, developers were "function point poor" compared to most other computer users.

In terms of paybacks, EDS refers to the Carnegie model: As investments are made in the development process, the paycheck will be repaid climbing from level 1 to 2. Paybacks from making the investment to get to a solid 3 or above are likely to be long term for virtually all organizations. [Note: According to studies INPUT is aware of, there is no strong correlation between Carnegie scores and identifiable payback.]

Hewlett-Packard

The following is HP's educated guess:

- In addition to whatever investment cost their might be in designing a new process or technique, the first time the process is put into use on an actual systems effort, there is probably a twenty percent penalty in development cost.
- On the second attempt the penalty probably drops to 10% and on future efforts gains in productivity can be made in the order of 10-20% per effort.

To pull together HP's C/S model probably cost between $500K and $1M. This doesn't count the many hours that were invested in developing stand alone components which were used in the model. However, benefits are there from the developers viewpoint.

After two years of working with the model in actual development efforts, they are probably delivering systems at 80% of the cost for these same systems developed using conventional methods.

IBM Corporation

The cost of making improvements can not be tracked on an overall basis in IBM and is not being tracked in most organizations. IBM is willing to make improvements to obtain near term business or to improve capabilities over a longer term. It is felt that many short term efforts should be undertaken in order to ensure that IBM will be aware of the more outstanding improvements.

Microsoft

The costs are significant, including technology and training. However, many of the costs are being absorbed as part of the development effort so are not really being measured.

The data that Microsoft has indicates that identifiable maintenance costs for C/S systems will probably be 50% of the original systems cost over a five year period. This compares to 70% for systems implemented under traditional architectures.

They have had some customers who feel that they got payoffs from applications implementations within weeks or months of implementation. Many of these applications were created within comparable timeframes.

Oracle

Unfortunately this is something that isn't measured. Partly because what is being done is intertwined with so many other activities, it's not clear that even a heavy investment in the measurement process would produce a result of significant value. However, major investments in new methodologies and tools were looked upon by users as long term investments. Only when the first applications came rolling out under the new approach would people begin to realize the potential savings or pay back.

Texas Instruments

TI thinks that changes in the systems building the are definitely a long term investment. They consider their overall program a 10 year initiative. This is particularly true in terms of dealing mainframe and legacy systems. However, they believe that they are already seeing benefits in terms of higher quality, shorter completion times and reduced maintenance costs. Systems developed under their new methodology appear to be yielding these benefits consistently and immediately. Their preliminary measurements under the new system showed reduction in maintenance and enhancement costs of 40%.

Xerox

No way to measure this. The major investment that was made by PARC/XSOFT to move to multiple platforms and open systems was analyzed and costs estimated, but they cannot disclose them. Since then, improvements in both the processes and products have been incremental in nature, and it would be impossible to determine the payback of these incremental changes.

From the point of view of payback, PARC/XSOFT look at the dollars they can pull in from the market. In that sense they feel that they have gotten rapid payback on investments through the migration of their offerings to multiple platforms. It's not clear how long it will take to get significant yield from the "open" strategy.

Although no scientific measure has been done, the changes in the development process which allow product to go to market more quickly with higher quality certainly improve the payback cycle.

Success/Failures

Borland

Some people might see their inability to bring the next release of Paradox to market earlier versus the release of Access by Microsoft as a failure. However, they believe they had fewer problems with their product. It is rock solid versus Access, and is now selling at a high rate.

Cambridge Technology

The use of the rapid prototyping technique to review system approaches and how systems will work has helped CT Partner's avoid real failures. Past experience has taught the organization that its methods must be adhered to in order to guaranty benefits.

Cap Gemini

CGA tries to avoid failures in using new systems building methods by working with business partners who are developing and testing out the methods.

DACOM

There were several major failures in business concept and execution in the mid-1980s.

- They were overly product oriented
- They did not sufficiently value or leverage their "wisdom" in the past from a business standpoint

Digital Equipment

There have been difficulties encountered, but failures have not been recorded or won't be discussed.

EDS

From EDS's viewpoint it has had great success with I-CASE. They can cite 10 to 15 implementations where EDS believes that the reduction in development time was in excess of 80% of the time it would have taken using conventional methods.

In terms of failures, EDS's biggest have come when user expectations weren't managed, and/or CASE capabilities oversold.

Hewlett-Packard

The C/S development model has been Hp's biggest success. On the failure side, early on they had an occasional disaster in terms of actually using the model to do a client implementation.

IBM Corporation

Failures or the inability to achieve total objectives has been notable in some CASE projects as well as projects run by IBM that were supposed to improve software building. A major lesson that was learned was that the business objectives and processes are sometimes neglected or not reviewed sufficiently and rethought during projects that automate systems building.

Microsoft

They would not comment on Microsoft failures except to say that these have been covered in the press, but were not fatal.

In customer implementations, the biggest "failures" have been in the management of user expectations. A second factor is the selection of "wrong" tools -- the tools may have been ok at the beginning but either new tools become available and/or there is a truer appreciation of what tools were needed. Both these factors will cause confusion in the market and increase the difficulty of measuring success.

Oracle

It's no secret that Oracle's biggest failure was in its initial approach to the applications software market. The products (originally developed for internal use) were low in quality and functionality, support was low quality, and sales commitments exceeded the deliverable capability. This problem is behind them now, and caused a major rethinking about the entire applications building process which forced Oracle into its new direction - with benefits all around.

Texas Instruments

Five major systems that have been built under the new systems methodology and are seen as successes. The corporation seems to be adopting the strategy rapidly and only with pockets of resistance.

TI feels that it is way ahead in business process engineering. They are already selling approach to certain business processes on a one-on-one basis. They anticipate that they will continue along these lines.

Xerox

Moving their entire product line from proprietary to open in under two years. (Success)

No comments on failures.

Other Organizations to Watch

Borland

The vendor they have to think about is Microsoft. They have been moving rapidly to work with software objects, but they have to support multiple processes.

It wasn't long ago that Bill Gates referred to objects as the same thing as subroutines. Now, his shop is really investing in object orientation. They can't match Borland's process as yet, in Borland's opinion.

Cambridge Technology

A number of systems integrators are improving their ability to build systems, but no one has taken the steps that CT Partners has to both speed up development and review a model of the business system before it is delivered.

Texas Instruments as well as Andersen are known for efforts to improve systems building.

Cap Gemini

Leading systems integrators such as big 6 firms and CSC are always trying to improve their ability to build systems. Andersen Consulting stands out as a firm that they believe has achieved improvements.

DACOM

James Martin Associates are doing good work. Otherwise, they admit to being inward-looking and do not "follow the competition". To some extent this is because they do not see most other vendors as being in the same kind of business as they are. That is, vendors that are working on software development improvements are not viewed as having many lessons for them. They view most other "business process re-engineering" vendors as being much more touchy-feely (or management consulting oriented). But they admit that they may be wrong.

Digital Equipment

These range from organizations concerned with improvements such as James Martin & Associates and KnowledgeWare to vendors such as Andersen Consulting and Cambridge Technology. Andersen has a methodology and tools that have been demonstrated on a number of projects.

EDS

Several organizations and institutions are seen as leading contributors to improvement in the systems building process.

- TI (IEF) as the leader in CASE technology and the first to seriously attach the issues related to new technologies.
- USC - Barry Boehm and the Software Engineering Institute for their developments in modeling and pointing the direction for ways to measure the software building process.
- The International Function Point User Group (IFPUG) for facilitating an industry wide standard which will likely become a solid measure of degree of difficulty and productivity.

EDS looks to Andersen and IBM as their primary competitors. They don't know for sure what they are doing, but believe both must be making advances on the same fronts as EDS.

However, EDS considers itself the leader. They've been working at transforming the systems building process from the top down for over four years; and have invested a large sum to get where they are.

Hewlett-Packard

HP is probably ahead of the average firm in rethinking and actually implementing changes to the software building process. They believe they were probably far from the head of the pack. They don't think that there was any one firm who was the leader in all aspects of the process.

IBM Corporation

The vendor mentioned most often as one who is or may be achieving such improvements is Microsoft. They have been building and rebuilding software objects so that they can be joined together more easily. Visible Basic has been improved and re-coded in C+ + to provide users the means of reviewing and adding to or modifying objects in a more easy fashion. The Next operating system of Jobs was felt to be a good step in the direction of improved development, but is not spoken of lately. Products such as Powerbuilder and the Insynch Passport system that generates C code are also being reviewed in addition to the products of TI, KnowledgeWare, Bachman and others whose products are under constant review.

Microsoft

They feel that Lotus has done good work in the applications and applications template areas (referring, essentially, to NOTES), but that Lotus had ceased being a leader.

They think Borland is poised to make large contributions. Borland doesn't have the leverage of Microsoft, but have done a lot of good thinking.

In terms of the "traditional" players, they think EDS is way ahead in its thinking in terms of utilizing C/S technology. DEC may be there soon also. SHL is also doing some interesting things in C/S applications integration.

Oracle

Some significant improvements are being made in the pharmaceutical industry. (Examples which confidential.) There are similar developments in the chemical where the adoption of new techniques and approaches to systems is allowing high levels of integration improving overall productivity to the firm.

Texas Instruments

In their industry TI believes that Hughes, Motorola and Raytheon have made significant progress (beyond TI's) in improvements to the systems building process. However, TI is ahead on business process re-engineering.

Xerox

No comments on other companies except to reiterate the concept that there are pockets of leadership all over the industry. From their viewpoint the PARC/XSOFT business relationship is unique and appears to be a good model. This may be an area where they are ahead of some of the rest of the pack.

Charging for Higher Value Software Assets

Borland

This is an issue that Borland is currently addressing. Products may be structured to facilitate adding capabilities. Then, changes can be made to add value.

Another means of obtaining more revenue would be to let the buyer chose components of the product he is interested in so that the buyer might only buy and pay for units a, b, d, and g of a product. Borland might offer a special deal if all components were bought at one time.

The product might also be produced in different versions for different markets. Charges could be different for each version, higher in the markets that could spend more.

Cambridge Technology

A systems integrator can obtain recognition and sufficient remuneration, particularly when steps such as the use of a rapid prototype are used. It might be hard to translate this process into building software products that commanded a premium price over other software products however since it could be hard to convince buyers that products offered a competitive advantage that would command a much higher price. Competitors would claim that they had something almost as good at a much lower price.

Cap Gemini

The total value of the assets can only be measured over time, although there may be more immediate benefits such as more rapid and less cost development or easier testing.

A systems integrator such as CGA can obtain recognition and remuneration based on the ability of delivering a solution. It might be necessary to establish a long term relationship to gain additional payment for delivering higher quality assets that would facilitate systems modification and growth for instance.

DACOM

They do not believe that their "consumers" are sufficiently educated to understand the issues involved. This goes back to the hard data question: If vendors (including themselves) could better demonstrate that this was true, then perhaps customers would pay more. They see this involving a long process of education and culture change -- 5 or 10 years at least -- before (a) This becomes true and (b) customers will accept it as being true.

Digital Equipment

Not sure how it can be measured and recognized, except over a period of time.

It will be necessary to guaranty the higher value in some way, if users are asked to pay more.

EDS

The asset benefit of software has got to be measured in terms of enterprise wealth. How this might actually happen presents a problem. In essence the software can only be measured in terms of added value. Evidence of this can be seen in a number of industries where mission critical applications are just now being attacked using new technology and methodologies. Examples:

- Transportation - Cargo Management
- Utilities - Customer Information Systems
- Telecommunications - Billing and Customer Information

Hewlett-Packard

HP doesn't feel that the asset value of individual custom designed systems will improve significantly as the direct result of improvements to the process; but does believe that the process will create new kinds of assets.

The business process models developed as a result of re-engineering can and (in a few instances) are being sold or brokered. Repositories of basic business processes will one day be sold. Using these models through a medium such as IEF, other firms could tailor both the processes and the code to meet their particular business needs. One example cited was an airline used TI to broker its IEF model for frequent flyer operations to another airline serving a non-competitive market.

IBM Corporation

Although it is believed that higher value software assets will be developed, there are unanswered questions about the ability of a vendor to achieve sufficient recognition for the software assets put in place so that a premium can be charged on a near term basis.

Microsoft

Software asset value will go up and up. The more software becomes integrated with individual business functions, the more it will be treated like "plant and facilities". Can anyone measure it? No! But Microsoft doesn't doubt that it's there. Over time there may be some way to measure it.

Oracle

The actual asset value of software will probably go down as it becomes easier to build and is replaced more quickly. However, the value of the "information asset" will likely rise significantly as newer applications and software systems permit its distribution, collection and utilization in ways that weren't previously possible. Whether anyone can measure these values is debatable.

Texas Instruments

The asset value of software can only be measured in terms of business value. Very little progress has been made with these kinds of metrics. The fact that they are able to sell on the outside some of the business re-engineering models shows that it is possible to get a measure.

Xerox

Software asset value will somehow have to be measured in terms of business contribution. The concept of software independent of the entire business process it drives has little or no asset value to the end user. The same software may have a great variety of "asset values", depending on how it's applied.

CHAPTER V:

Summary By Vendor

Borland

1. Changes in the Software Building Process (Overall)

There are three reasons why Borland is changing from procedural to object oriented development.

- The software that results is more reliable; it takes less time to debug and fix code.
- The code that results is more suited to being reused.
- It is easier to make revisions and enhancements in the future.

In effect, the time and cost required to modify software can be reduced. Other benefits such as the reduction of elapsed time for a project or cost of developing new software has not or can not be quantified.

2. Soft Benefits

Object orientation forces developers to do more up front planning in order to avoid excessive complexity.

Object orientation provides a more powerful type of development in which developers can work with a reasonable number of blocks rather than with millions of lines of coding - which has to reduce the complexity and time of development.

3. Relative Contribution of Tools and Processes

Borland has focused more on the process than on tools. The use of tools, including C++, Object Windows and libraries, is important in relation to support the process.

The process involves using a library of objects, tools to help modify old objects and develop new ones and tools for linking objects. A GUI is used with the process and it has aids to ensure good practices are followed but high level experience/skills are needed.

They can also add other tools such as expert systems that will allow developers to check off boxes as part of the process.

4. Reusability

One of the objectives of Borland's object oriented process is to develop reusable objects. During the development of one software product, they develop or modify objects that would be reusable for other products.

5. Difficulties in Addressing Client/Server

Difficulties involved in separating processes or splitting the management of data can occur and require planning that might, for instance, involve the use of a common data base. These difficulties are addressed in internal development efforts.

Borland also has a product, Interbase, now in use in the Philadelphia stock exchange, that addresses the need for a database that can be used by multiple systems. Borland referred to Interbase as middleware.

Borland sees itself as a client/server oriented firm.

6. Is More Than One Process Model to Improve Systems Building?

Borland is now dedicated to the use of one process model, the object oriented model of systems building. Change is managed or coordinated on an overall basis to a great extent (4.0 on a scale of 1 to 5 where 5 = high level of management control).

7. Skill Needs and Distribution

Skills to support the process carried out at Borland are in short supply. Proactive steps to improve the situation include careful hiring practices with some universities as well as internal training. Programmers and QA people will need 3 to 6 months of internal training to become useful.

New recruits are sought at a handful of universities, who do things the "right way" then, the recruits are given high level training for three to six months. The result may be a creation/selection of "prima donnas".

8. System Testing Process

It is evolving to become more modular.

Testing for a given type of an engine, for instance, could always be handled by the same team. This would be similar in concept to having packaging and marketing handled by specialists.

9. Particular Innovations

The object oriented process they use is particularly innovative.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

There was a cost in terms of the delays encountered in bringing certain products to market (Quattro and dBase releases). They feel, what they accomplished will pay off in the long term since the products are more solid, less in need of fixes.

11. Success/Failures

Some people might see their inability to bring the next release of Paradox to market earlier versus the release of Access by Microsoft as a failure. However, they believe they had fewer problems with their product. It is rock solid versus Access, and is now selling at a high rate.

12. Other Organizations to Watch

The vendor they have to think about is Microsoft. They have been moving rapidly to work with software objects, but they have to support multiple processes.

It wasn't long ago that Bill Gates referred to objects as the same thing as subroutines. Now, his shop is really investing in object orientation. They can't match Borland's process as yet, in Borland's opinion.

13. Charging for Higher Value Software Assets

This is an issue that Borland is currently addressing. Products may be structured to facilitate adding capabilities. Then, changes can be made to add value.

Another means of obtaining more revenue would be to let the buyer chose components of the product he is interested in so that the buyer might only buy and pay for units a, b, d, and g of a product. Borland might offer a special deal if all components were bought at one time.

The product might also be produced in different versions for different markets. Charges could be different for each version, higher in the markets that could spend more.

Cambridge Technology Partners

1. Changes in the Software Building Process (Overall)

They have introduced changes to bring about more rapid development and will add to or refine them in the future. They have speeded up the development process or time to develop an important or strategic application system by a factor of 3 (time to completion and use of the system) compared to similar projects at a client or other company where traditional methodologies or even some newer methods were used. (This appears to be so, but is based on using people with considerable knowledge and experience employed who impose a methodology that forces higher level user executives to provide timely and meaningful contributions to the process.)

The process allows the cost of the software building process to be reduced (including the person months involved) and the quality to be increased.

The development technique includes a well planned and executed preliminary requirements definition, rapid systems prototyping methodology to pin down requirements and parallel development teams to implement the system. (Top user executives are forced to provide timely and dedicated reviews of the results of the prototype through the activities of CT.)

2. Soft Benefits

The use of rapid prototyping during the requirements definition helps to improve communication about requirements as well as to analyze what is going on in business processes. If business systems are being re-engineered, concepts can be tested out. The prototype software enables users to review systems and gives them a feeling of satisfaction with the final implementation.

3. Relative Contribution of Tools and Processes

Both tools and processes are used in the CT Partners methodology. The rapid prototyping capability is more of a tool-like technique that allows concepts to be tested.

The development of the implementation code by the use of parallel teams is more of a process or factory like method in which one team handles the application, a second handles connectivity, a third the user interface, and a fourth the data.

4. Reusability

The methodology described above has not focused on reusability, but the use of parallel development efforts would facilitate the consideration of reusable code since functional aspects of coding are separated.

5. Difficulties in Addressing Client/Server

There can be difficulties since corporate objectives as well as processes and data can be divided between uncoordinated systems on client/servers. Cambridge Technology allows client/server systems to be handled by themselves or better yet, together with other, related systems changes that are necessary to meet corporate goals. The methodology utilizes open systems which facilitates work with a variety of client/server technology.

They also test out application approaches with rapid prototyping that should highlight questions about the segmentation of functions and data across systems.

6. Is More Than One Process Model to Improve Systems Building?

CT Partners uses only one process model, although improvements to it are considered, evaluated and implemented over time. They have a high degree of control over management of change.

7. Skill Needs and Distribution

Higher level skills are in short supply and distributed unevenly. CT Partners provides a method that allows the need for skills to be managed and controlled more easily.

The CT Group, an affiliated organization, provides education in the methodology utilized by CT Partners which provides more opportunity for overcoming skill deficiencies.

8. System Testing Process

Users are being brought in to systems testing more, particularly to review the systems concept or business re-engineering envisioned. The methodology of CT Partners is supportive of this by enabling users to formulate requirements in a structured manner and test out the system approach more quickly through the rapid generation of a prototype.

9. Particular Innovations

CT Partner's believes its methodology (described in section 1) is particularly innovative.

CT Partner's also feels that its ability to rapidly develop strategic systems using an open system approach is innovative.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

Cambridge Technology has experience in relation to this topic. It originally sold education that concentrated on using technology to obtain competitive advantage. The cost of implementing the methods it advocated were hard to justify in many organizations. It was easier for these organizations to pay for the use of the methods to implement strategic applications more rapidly. A near term payoff on a project justified using the methods.

Since the current process is so controlled, the cost of making improvements to it can be evaluated in terms of both near and longer term benefits. No changes are foreseen that would change the process to a significant extent.

11. Success/Failures

The use of the rapid prototyping technique to review system approaches and how systems will work has helped CT Partner's avoid real failures. Past experience has taught the organization that its methods must be adhered to in order to guaranty benefits.

12. Other Organizations to Watch

A number of systems integrators are improving their ability to build systems, but no one has taken the steps that CT Partners has to both speed up development and review a model of the business system before it is delivered.

Texas Instruments as well as Andersen are known for efforts to improve systems building.

13. Charging for Higher Value Software Assets

A systems integrator can obtain recognition and sufficient remuneration, particularly when steps such as the use of a rapid prototype are used. It might be hard to translate this process into building software products that commanded a premium price over other software products however since it could be hard to convince buyers that products offered a competitive advantage that would command a much higher price. Competitors would claim that they had something almost as good at a much lower price.

CAP Gemini

1. Changes in the Software Building Process (Overall)

They have ongoing programs concerned with improving the software building process, although their main concern is satisfying the needs of clients in specifying, delivering and installing business systems. They focus attention about improvements in the software building process on activities of software engineering firms such as Intersolv and Bachman who have been business partners.

Changes in the process which would speed up the development of business solutions, increase quality and/or reduce costs could increase competitiveness and be important to us. Improvements are being obtained on a regular basis, but they would like to see a substantial or notable change that could reduce the time for large, complex projects by 25% or more. This would gain attention from clients and prospects. However, control of change to their process is most important.

2. Soft Benefits

Improvements in software building should include steps to aid users to review how business processes will be changed or re-engineered as well as to review data management and utilization. Improvements in software engineering tools have aided both of these objectives.

3. Relative Contribution of Tools and Processes

As noted above, tools have brought ongoing improvements in the software building process, but changes in the overall process have made the greatest impact on their ability to deliver timely and quality work, and we would like to see an improvement in the process that would provide a substantial improvement in time and cost requirements while also enhancing quality further.

They are reviewing the use of factory-like methods to generate code although most of the effort is devoted to working with users and difficulties could occur in trying to implement a factory process in that situation.

4. Reusability

Cap Gemini and its business partners are exploring the use of reusability of objects as well as object oriented programming, but this is one of many efforts to improve methods.

5. Difficulties in Addressing Client/Server

The needs of users have to be fully understood before decisions are made regarding the technology that will be used in a solution. Decisions are made too quickly in regard to the use of client/server technology, and problems can result in relation to data management and the division of functions between business units.

6. Is More Than One Process Model to Improve Systems Building?

Within CGA, there is an effort to use one model although experimentation is made with other models. We are willing to support projects at users that employ models different from the one we support internally. Some individual offices or personnel are attempting to change methods, but control is achieved. They have a medium-high degree of control over the management of change.

Due to uncertainties about the systems building process as well as the lack of coordination between various department in a company, multiple models may be in use at a client, and they might be involved in supporting more than one.

7. Skill Needs and Distribution

There are currently needs for more software development skills due to the variety of tools, languages, databases, operating systems, and other systems and application software in use for workstations as well as other platforms. CGA as well as other SI and professional services firms can not count on having the skills available that prospects and clients have in use or plan to utilize. Alliances, temporary staff members and the services of specialized temporary employment services must be used.

Skills bottlenecks are being encountered. CGA will propose changes in the software products planned or specified for jobs by users. Recently, one client was persuaded to step back to an earlier version of a workstation operating system in order to use the network manager which had been selected.

In addition to proposing changes to the set of software products selected for a project, the proactive step more often taken by CGA is to work with clients and recommend a set of products that can be well supported.

Internal education as well as relations with consultants and other firms are also sought to expand the skills inventory that CGA can support.

8. System Testing Process

More comprehensive tests are formulated with the aid of users, and users are brought into the review of proposed systems at an earlier stage through the use of front end CASE tools and JAD or RAD types of processes.

9. Particular Innovations

The effort to use the software engineering tools and concepts of leading software engineering firms together with the experience of CGA is felt to be innovative. Some other SI/professional services firms concentrate on using their own tools and methodologies. CGA wants to incorporate the ideas of firms who specialize in improving the software building process.

Position papers and internal training ensures that innovations are spread throughout CGA. CGA is prepared to work with clients to help them speed innovative changes in systems building, but this can be difficult to achieve in large organizations, since many user groups are acting independently.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

Improvements in systems building within CGA takes time and consume funds and personnel, and can delay getting work done and limit or preclude taking advantage of business opportunities. That is why CGA tries to use the experience of business partners active in systems development processes and tools as well as experiment with methods that might improve systems development. CGA wants to be able to work with clients that want to use various improvements, even if temporary consultants are employed to aid with that; however, CGA wants to limit internal changes unless there are longer term advantages. .

11. Success/Failures

CGA tries to avoid failures in using new systems building methods by working with business partners who are developing and testing out the methods.

12. Other Organizations to Watch

Leading systems integrators such as big 6 firms and CSC are always trying to improve their ability to build systems. Andersen Consulting stands out as a firm that they believe has achieved improvements.

13. Charging for Higher Value Software Assets

The total value of the assets can only be measured over time, although there may be more immediate benefits such as more rapid and less cost development or easier testing.

A systems integrator such as CGA can obtain recognition and remuneration based on the ability of delivering a solution. It might be necessary to establish a long term relationship to gain additional payment for delivering higher quality assets that would facilitate systems modification and growth for instance.

D. APPLETON COMPANY (DACOM)

1. Changes in the Software Building Process (Overall)

The current approaches, especially CASE, haven't just reached a dead end but have lead us down the wrong dead end: The problem isn't producing code efficiently and correctly, but designing the right systems in the first place.

DACOM began as a CASE-oriented, essentially product, company in the early 1980s. It built reasonably good tools and was (and is) a leader in repository thinking. However, over time DACOM saw that the chief need was for an understanding of business processes and making improvements in the systems which support business processes.

DACOM has evolved from a company with an emphasis on software building to one that works closely with clients to design the right system in the first place. Software "implementation" is viewed by DACOM as almost easy once the underlying business design is set.

Overall time to complete may or may not be shorter compared to conventional methods. Much depends on how well their clients have been educated in the DACOM method. Initial projects are usually slower because of education and cultural change. Subsequent projects are probably shorter; however, this is very difficult to measure since historical measures of time to complete don't do justice to the time spent on the up-front business engineering phase.

Actual software development is much faster after DACOM's work:

- There are relatively few downstream changes
- The "spaces" are unambiguous

Quality is immeasurably higher, although, again, there are few benchmarks and measuring methodologies for quality. This is very frustrating for DACOM because they and their clients strongly believe this to be true, but it is very difficult to prove. Other vendors make similar claims -- which DACOM doubts are true -- and DACOM's real achievements (to them) get lost in the general noise level.

DACOM admits that they could do a much better job at quantifying benefits, in order to

- Measure their own progress
- Assess the relative strengths of different approaches
- Market themselves better

Right now, too much of their acceptance is based on clients buying into the DACOM "religion". This is ironic and frustrating for an organization that views itself as a rigorous, engineering-oriented company.

2. Soft Benefits

DACOM would take issue with the concept of "softer" benefits. The benefits are softer only from the standpoint that many (maybe most) of the benefits do not accrue to software building, but are gained by the underlying business. (DACOM agrees that it would be very powerful to measure the "before" and "after" business costs and benefits, but it does not do so on a consistent basis. Its clients don't want to pay for this kind of activity, because doing so won't help solve their particular business problem.}

DACOM works largely with true end users in the manufacturing/logistics area. From the standpoint of DACOM's clients the current system building process is "intolerable" and not just for the usual reasons of lateness and expense. The more thoughtful clients (and DACOM appeals to thoughtful clients, for better or worse) are frustrated by software that

- Doesn't meet their business functional needs very well.
- Behaves in inconsistent and unpredictable ways; users are overwhelmed by system details; computer systems make it harder rather than easier to understand underlying problems.
- Is difficult to change in the terms of time and expense and, even then, often can't really be changed to meet user needs

Downsizing of organizations and of the computing environment multiplies these problems

Their clients don't really care about the usual measures used to measure a "good" software project.

3. Relative Contribution of Tools and Processes

DACOM has a full perspective, having started as a CASE company (before the term was invented) and now describe themselves as a "systems integration consultant."

DACOM does use tools, some of which are their own (see attachment 1).

However, they see tools as essentially supporting a model-oriented approach ("Tools are the finger painting of business engineering").

- They see almost all CASE tools as not supporting the improvement of business processes (although pieces of them can do so with the right orientation)
- A good repository is vital, but they don't know of one that really meets their needs.
- They are forced to use and link standalone tools. They hope that in the future there will be integrated tools that will support a range of modeling techniques.

A key component (and bottleneck) is the need for consultants with the necessary "wisdom" to integrate the business problem, modeling techniques and client change. This is the strength of their business, but also a key limitation.

- These traits make them valuable to customers, but expensive.
- DACOM would like to be able to leverage this knowledge in a more embedded methodology. They see the application of AI and fuzzy logic as possibilities, but are currently resource-constrained in being able to pursue these directions very aggressively. (They are especially cautious since in the past they have tried to prematurely productize processes.)

The general process that they use is shown in Attachment 2.

4. Reusability

They are attracted to the concept of reusability but have found only a small number of "paradigm processes" which they can see as potential candidates. For example, "inventory" should qualify, but doesn't once the major differences between industries are examined as well as the "flavors" between firms.

They see no theory of reusability in the commercial world (unlike the formulas in the scientific world) to make their job easier. They use common skeletons and analyze differences but don't really view this as reusability.

They reuse "wisdom" and are aware of the deficiencies in doing so.

5. Difficulties in Addressing Client/Server

The client/server model is extremely incomplete. More basically, there is no model for organizational or functional downsizing. This puts a process engineering in a double bind. Learning for DACOM is "painful" and largely intuitive as far as where data, processing and location boundaries are drawn.

This disturbs them less than may appear, since the analytic process involved in understanding the business processes will usually keep them from making unrecoverable mistakes in "client" and "server" function allocations.

They feel great sympathy with their partners (whether customer-partners or vendor-partners) who are trying to make client/server technology work.

6. Is More Than One Process Model To Improve Systems Building?

They have their own approach (attachment 2). However, they feel agnostic toward particular business analysis practices and feel it is too early to back one horse. They have a high degree of control over the management of change.

7. Skill Needs and Distribution

Finding people who understand business processes is their greatest need. It takes several years to train a new employee with little business experience (largely on the job). They can predict fairly well the people who will not ultimately make it, after a day of pre-qualification testing; however, they can only predict those who will not fail; they are much less sure of those who will do very well. They have often been forced to hire more senior people as a result; this kind of person is more expensive, pound for pound, than a more junior person. A larger problem is that it is more difficult to form a cohesive "DACOM way" from people who are more set in their ways (even if these are sound, productive ways).

DACOM would like to emulate the philosophy of an EDS or Andersen and mold new recruits to their culture.

8. System Testing Process

Testing should be done at the model level.

9. Particular Innovations

DACOM believes that their entire approach is innovative, i.e.,

- Emphasizing modeling the upstream business process and defining objectives
- Focus on model building
- Making customer part of team, involving high levels of education and training
- De-emphasize conventional "software building" part of the equation (even though many of the DACOM staff has a heavy background in software building).

10. Cost of Making Improvements in Systems Building (Funds, People, Opportunities Foregone)

High costs, both to themselves and their customers. [A strength and weakness of DACOM is the fuzzy line between themselves and their customers.] DACOM has plowed back a great deal of time and money into iterations of their approach. They believe that they are on a very productive path now. But the whole process has definitely been a long term investment for them.

11. Success/Failures

There were several major failures in business concept and execution in the mid-1980s.

- They were overly product oriented
- They did not sufficiently value or leverage their "wisdom" in the past from a business standpoint

12. Other Organizations to Watch

James Martin Associates are doing good work. Otherwise, they admit to being inward-looking and do not "follow the competition". To some extent this is because they do not see most other vendors as being in the same kind of business as they are. That is, vendors that are working on software development improvements are not viewed as having many lessons for them. They view most other "business process re-eningeering" vendors as being much more touchy-feely (or management consulting oriented). But they admit that they may be wrong.

13. Charging for Higher Value Software Assets

They do not believe that their "consumers" are sufficiently educated to understand the issues involved. This goes back to the hard data question: If vendors (including themselves) could better demonstrate that this was true, then perhaps customers would pay more. They see this involving a long process of education and culture change -- 5 or 10 years at least -- before (a) This becomes true and (b) customers will accept it as being true.

Attachment 1

Also From D. Appleton Company:

D. Appleton Company is the premier business engineering consulting firm offering business process improvement consulting, publications, workshops, seminars, and software tools to support both the government and commercial environments.

Listed below is a sample of the products, services, and training D. Appleton Company offers in support of business engineering. If you are interested in any of the following offerings or would like to receive more information about D. Appleton Company, please call us toll-free at (800) 322-6614.

Publications

- Newsletter: *Business Engineering Newsletter*
- Guidebook: *Corporate Information Management Process Improvement Methodology For DoD Functional Managers*

Workshops & Seminars

- Advanced Data Modeling
- Business Benchmarking
- Business Engineering
- Enterprise Integration
- IDEF Modeling
- Information Asset Management
- Logical to Physical Design
- Modeling for Managers
- Process and Data Modeling Techniques
- Process Improvement Methodology Workshops
- Prototype Development Methodology
- Quality Function Deployment (QFD)
- Requirements Analysis Planning
- Team Techniques

Software

- Demonstration Disk of DoD Process Improvement Methodology Support Tools
- ModelPro™ — Windows- and Macintosh™-based IDEF1X Data and Rule Modeling
- Design/IDEF™ — PC- and Macintosh-based IDEF0 Process and Activity Modeling
- EasyABC™ — DOS- and Macintosh-based Activity-Based Costing
- IDEF/LEVERAGE® — MVS- and VMS-based IDEF0 and IDEF1X Modeling and Database Design.

Copyright by D. Appleton Company, Inc. ModelPro and IDEF/LEVERAGE are trademarks of D. Appleton Company, Inc. All other software product references are trademarks or registered trademarks of their respective owners.

S66

Call today for more information
1-800-322-6614

225 S. Sepulveda Blvd., Suite 300, Manhattan Beach, CA 90266, 310/374-3939 FAX 310/374-2041
222 W. Las Colinas Blvd., Suite 1141, Irving, TX 75039, 214/869-1066 FAX 214/869-1099
3028 Javier Road, Suite 400, Fairfax, VA 22031, 703/573-7644 FAX 703/698-8219
4 Broadway, Suite 9, P.O. Box 206, Valhalla, NY 10595, 914/682-5955 FAX 914/682-5945

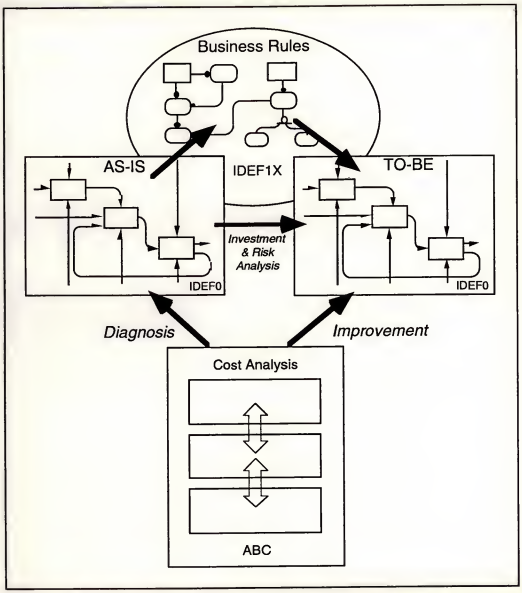


Figure 8-1: Role of Business Process Improvement Techniques

Digital Equipment Corporation

1. Changes in the Software Building Process (Overall)

Changes are being considered or made to speed up the development of software as well as to reduce its cost and improve quality. Many separate activities are taking place ranging from steps in consulting and systems planning to work at all levels with tools and methodologies. Specific goals have not been quantified or can't be discussed, but it seems that the speedup being sought at this time would reduce development time by one third on larger projects.

2. Soft Benefits

There is a definite need to achieve other, softer benefits for customers including improved satisfaction with the development process, improved communications on the work being done, and more ability to respond to needs. These needs are being investigated in relation to systems integration and professional services work as well as in relation to internal analyses of systems building.

3. Relative Contribution of Tools and Processes

Tools including CASE tools and other software aids are being sought and evaluated to aid with the system building process. Tools may seem to be more important in regard to current improvements in systems building, but improvement of the entire process that is carried out in systems building offers more opportunities for improvement.

The use of techniques such as JAD or RAD as part of a process are expected to have more impact on time, cost and quality than tools can.

Research is being conducted that addresses both the use of tools and processes to improve the system building process. The use of manufacturing techniques in software production has been carried out and is also being investigated together with other topics.

4. Reusability

Tools are being used that allow a repeatable solution to be sold. For certain applications, these tools will allow the solution to be modified or customized more easily, and many components of the solution would not have to be changed. There have been various types of reusability in the past by DEC VARs as well as by DEC. Digital has also been experimenting with reusing objects in a group of internal projects and is now doing that on a regular basis

5. Difficulties in Addressing Client/Server

There are questions about the allocation of data and processes from a consolidated system to a number of client/server systems. There are also questions about using a set of data or transactions received at one point to update multiple client/server systems.

6. Is More Than One Process Model to Improve Systems Building?

DEC is using multiple process models, but most of these efforts are coordinated. In some circumstances, activities in support of accounts could use tools and techniques or new methodologies in an unplanned manner to improve systems building. Where a VAR or associated vendor is in a lead position, this is particularly true. They have a medium degree o control over the management of change.

Tools are available to help manage or control change management at DEC, but they are not always used.

7. Skill Needs and Distribution

There are a number of new skills that are needed in current projects ranging from those needed to use new systems development and relational database tools to those available to use network software products, GUI and workstation spreadsheet and database products.

The new skills are not always present at DEC. Some internal training is carried on and external training is made use of. However, a group of contract services vendors are used to meet critical situations with temporary people. VARs and associated consultants are also used to meet critical needs.

8. System Testing Process

Efforts are being made at Digital to develop new types of testing that will be automatically incorporated into the system building process.

There are also some products being introduced jointly with large companies that will handle portions of the testing process in new ways through system review with front end CASE tools, prototyping and other means.

9. Particular Innovations

Digital feels that its system building process is innovative since it results in complex network solutions in a relatively short time.

Several recent internal efforts have resulted in factory-like development of code, but can't be discussed yet.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

It is hoped that there will be some rapid or near term improvements although most observers at DEC feel that it will be a long term investment.

11. Success/Failures

There have been difficulties encountered, but failures have not been recorded or won't be discussed.

12. Other Organizations to Watch

These range from organizations concerned with improvements such as James Martin & Associates and KnowledgeWare to vendors such as Andersen Consulting and Cambridge Technology. Andersen has a methodology and tools that have been demonstrated on a number of projects.

13. Charging for Higher Value Software Assets

Not sure how it can be measured and recognized, except over a period of time.

It will be necessary to guaranty the higher value in some way, if users are asked to pay more.

EDS

1. Changes in the Software Building Process (Overall)

They see very few changes in the "old paradigm" i.e. maintenance and minor improvements to existing legacy systems. However, they believe that for new development the changes are dramatic, both in methods and technology. The shift is from monolithic systems designed in the IBM model to C/S and open.

- "In the IBM model we forced the system to meet the capabilities of the technology."
- "In the new model we mold the technology to the system requirements."

The new approach takes an enterprise view with much more engineering orientation. The old was more artistic. They are firm believers in the 5 level Carnegie model of software development capability maturity: (1) Ad Hoc, (2) .., (3).., (4).., (5) Optimizing phase where the manager doesn't worry about the quality of the code, but rather adherence to the process. EDS feels that most commercial organizations are operating at level 1 or 2 on the scale.

Three other significant developments which are changing the way software is built are:

- Away from the total "top down" waterfall approach, from high level design to detailed implementation to "spiral" approaches which iterate to a solution. In essence, the RAD (Rapid Application Development) approach.
- The use of JAD (Joint Application Development) with users and developers sharing responsibilities throughout the development process.
- "Time Boxing" - an approach where a fixed amount of time is allowed for specification development. At the end of that period, implementation is done putting usable functionality into place. The process is then repeated on the existing system to obtain higher levels of functional refinement

Benefits Include:

- Productivity

  EDS measures productivity through the use of function points. They see this formerly proprietary approach maturing into a set of industry standards through the International Function Point User Group (IFPUG). EDS is now bidding some jobs on the basis of function points. Utilizing the new methodologies coupled with I-CASE where appropriate, they claim that EDS can get significant improvements in terms of function points implemented per hour compared to traditional methods.

- Shorter Time to Complete

  Can complete discrete applications (at the departmental level) in 70% of the time that it use to take to get to operational status. The reduction in time to complete is less for full applications suites, or enterprise wide systems. The methodology described in the attachment will begin to attack that problem.

- Quality

  Faultless code - increased functionality. EDS has implemented a standard metrics system (architecture for capturing metrics) for getting at the quality measurement problem, but it hasn't been in place long enough so that they can make any definitive statements regarding quality comparisons based on methodology deployed.

- Reusability

  Where CASE tools are applicable to the development being undertaken, a fair amount of reusability can be anticipated. However, most CASE tools are far behind in dealing with interface development, objects, and multimedia, all of which are becoming an increasing part of systems requirements.

  They are trying to develop a corporate data base on quantification of benefits, but the number of efforts which have been submitted to the metrics data base is small. When they get this into full production, they anticipate that there will be different levels of benefits - from cost saving to enterprise-wide benefits. EDS has shifted its thinking from its former view (primarily as an outsourcing vendor) of optimizing operations, to optimizing the development process. This is where they feel the real benefits will come.

2. Soft Benefits

"The hard stuff is easy, the software stuff is hard. Soft stuff is harder to get than the hard stuff." Quote aside, they feel that the biggest benefit on the soft side is the benefit obtained through the change management aspects of new systems implementation methodologies: i.e. if done properly major new implementations transform the business process, either directly or indirectly.

3. Relative Contribution of Tools and Processes

- Process and organization

  Process is the key and the tools must fit the process. Organizations need to focus on measuring the process...not the people. They believe this is the biggest failure of most implementation processes. They also believe that much progress is being made in the "manufacturing" approach to systems implementation, but it goes in "fits and starts" depending upon the availability of the right tools to match the process.

- Tools

  From a CASE standpoint, EDS uses both IEF and its own product INCASE. Other tools exist to manage the overall process as described in the attachment. In general however, all these tools need to be updated significantly to deal with today's architectures and development processes.

  In terms of AI and embedded methodologies, EDS believes that a lot is already there, and expects to see more and more "intelligence" and "judgment" built into tools as the evolve.

4. Reusability

There are varying degrees of reusability from program elements, routines, shells and now objects. Even though EDS is making heavy investments in object technology, there's a long way to go. The primary problem is knowing what's in the object library and precisely what it does. In other words the conceptual approaches to object management have a long way to go. An EDS saying is that most object library systems are like a roach motel...a lot check in, but damn few are heard from again!

The most reusable parts of most commercial systems are likely to be the data models. They've had great success in this area. In other words, processes may vary from one firm to another, but the fundamental data elements and their relationships to each other in most operational systems are the same, or close enough, that significant productivity can be gained by migrating data models between like applications.

5. Difficulties in Addressing Client/Server

Yes, the client/server model is incomplete particularly at the high level; i.e. there are very few universal models for the design phase of client/server applications; i.e. allocation of processes and data between the client and server. This is particularly true where portability across multiple architectures is a system requirement.

However, EDS believes that the CASE tools are evolving to deal with these issues on a static basis. The real long term objective however, should be to design C/S systems which can reallocate functionality and data between the client and the server dynamically, based on transaction volume, or other key systems parameters.

EDS uses SES (Scientific Engineering Software Model) which helps designers forecast performance at the design phase as a function of the allocation of data and processes and transaction loading models.

6. Is More Than One Process Model to Improve Systems Building?

Yes, a single one won't work! In fact EDS's entire development process is based on that assumption. See attachment, "EDS Methodology Architecture". The underlying assumption is that you can force an overall architecture of framework, but the framework must be able to accomodate some variations in methodology and even more in some specific processes. Within this context, change is managed/coordinated quite well. (4.0 on a scale of 1 to 5, where 5 = high level of management control.)

7. Skill Needs and Distribution

There is enormous change in this area driven primarily by three factors:

- Demands for new kinds of applications (new processes to automate) including business support, new kinds of decision support, etc.

- New techonology. In the old IBM scenario, the bottom line was that you were going to use a 3270. This is totally unacceptable today. The choices of hardware architecture alone present serious challenges. EDS has calculated that using just the most popular hardware and software products that the number of combinations of platforms for a typical C/S application could theoretically approach 7.6 times 10 to the 50th power. So the trick is to isolate the development process from the platform. New interfaces and new media also present a challenge on the training and skill development front.

- New processes of systems building.

EDS is focusing on shifting its training emphasis to accommodate these new requirements in a revolutionized environment. They are placing heavy emphasis on interface design I-CASE, C++, object development techniques and 3GL languages. And they continue to expand their Interactive Distance Learning Network (IDLN). This provides a virtual classroom environment where professionals can lock in from anywhere in the world, and using interactive keypads, participate in classroom sessions. Instructors can obtain instant feedback, etc.

Right now this capability is limited pretty much to teaching technical skills, but that new courseware is being developed which will allow people to learn about processes and higher level concepts as well.

8. System Testing Process

EDS currently has little to offer here. They believe that a lot more emphasis is being and will continue to be, placed on testing at every phase of the life cycle. The use of I-CASE has simplified the testing process and reduced the number of bugs which must be fixed significantly. The less code you write by hand, the higher the quality. On many implementations 90% of the code will come through CASE and the remaining 10% will be written by hand. (Of course, the 10% is more difficult).

The other thing that's changing in the testing process, is the heavy involovement of users in all aspects of testing.

9. Particular Innovations

EDS feels its particular innovations include:

- Inference based code generation (CDM)
- Rapid interactive systems engineering (RISE)
- Electronic distribution of life cycle, courseware, and processes

EDS's approach to an overall framework as described in the attachment is also rather innovative and unique.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

The costs are high. Excluding retraining in technology and processes, the cost of properly outfitting a developer to operate effectively in the new environment is $70,000; $50,000 for hardware and software tools, $20,000 for "objects" and other development aids. EDS is a firm believer that this should be done, and believes that up until recently, developers were "function point poor" compared to most other computer users.

In terms of paybacks, EDS refers to the Carnegie model: As investments are made in the development process, the paycheck will be repaid climbing from level 1 to 2. Paybacks from making the investment to get to a solid 3 or above are likely to be long term for virtually all organizations. [Note: According to studies INPUT is aware of, there is no strong correlation between Carnegie scores and identifiable payback.]

11. Success/Failures

From EDS's viewpoint it has had great success with I-CASE. They can cite 10 to 15 implementations where EDS believes that the reduction in development time was in excess of 80% of the time it would have taken using conventional methods.

In terms of failures, EDS's biggest have come when user expectations weren't managed, and/or CASE capabilities oversold.

12. Other Organizations to Watch

Several organizations and institutions are seen as leading contributors to improvement in the systems building process.

- TI (IEF) as the leader in CASE technology and the first to seriously attach the issues related to new technologies.
- USC - Barry Boehm and the Software Engineering Institute for their developments in modeling and pointing the direction for ways to measure the software building process.
- The International Function Point User Group (IFPUG) for facilitating an industry wide standard which will likely become a solid measure of degree of difficulty and productivity.

EDS looks to Andersen and IBM as their primary competitors. They don't know for sure what they are doing, but believe both must be making advances on the same fronts as EDS.

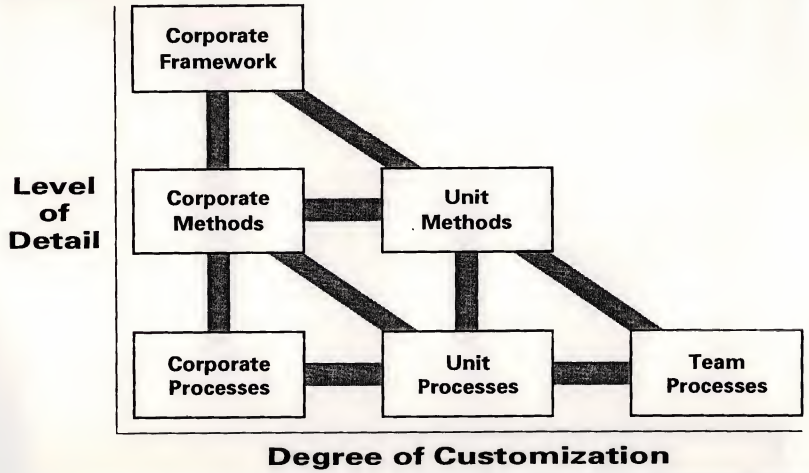
However, EDS considers itself the leader. They've been working at transforming the systems building process from the top down for over four years; and have invested a large sum to get where they are.

13. Charging for Higher Value Software Assets

The asset benefit of software has got to be measured in terms of enterprise wealth. How this might actually happen presents a problem. In essence the software can only be measured in terms of added value. Evidence of this can be seen in a number of industries where mission critical applications are just now being attacked using new technology and methodologies. Examples:

- Transportation - Cargo Management
- Utilities - Customer Information Systems
- Telecommunications - Billing and Customer Information

EDS Methodology Architecture



EDS Methodologies

EDS has developed a corporate methodology architecture, which serves as the foundation for continuous improvement of its processes and as the source of leverage for corporate education and training. Processes supporting the lines of business, defined by the EDS Service Continuum, should be based on this architecture.

EDS methodologies are represented by a two-dimensional architecture, shown in Figure 3.

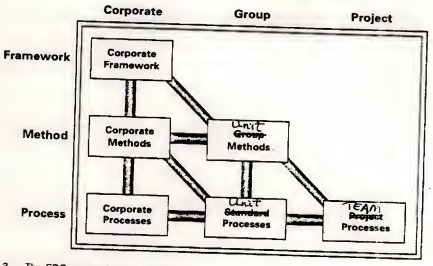


Figure 3 — The EDS approach to methodologies

The vertical dimension identifies levels of detail and precision and is divided into Framework, Method, and Process:

- The Framework level of a methodology defines the essential components that must be addressed to effectively support an EDS line of business. It links together the levels of detail and provides the common vocabulary and rules for all methods that support that line of business. For example, the Systems Life Cycle (SLC) framework relates the supporting disciplines of Project Management, Metrics, Configuration Management, and Requirements Determination to classical elements, such as Analysis, Design, and Construction.
- The Method level describes the linkages among the framework components. A method specifies the content, relationship, and purpose of individual tasks and work products. In addition, it may reference specific techniques for accomplishing these tasks, and it may constrain the choice of tools. For example, SLC-I is the name of the corporate method that employs Consolidated Data Modeling and is supported by INCASE. Group methods may be developed within the methodology framework for situations in which a group's business addresses a niche market or a new technology. These methods may later be adopted and supported at the corporate level. Both corporate and group methods are valuable because they provide a rigorous foundation for building work processes.

- The Process level describes the detailed steps that constitute actual execution. Processes are based on specific types of work and integrate many components required for successful project completion. A method and associated tools and techniques must be chosen, applicable tasks must be selected, and standards must be incorporated. In addition, supporting disciplines defined in the framework must be integrated in sufficient detail to provide repeatability across multiple projects.

The horizontal dimension identifies ownership and customization and is divided into Corporate, Group, and Project levels. In this context, "group" refers to a collection of organizations that have decided to do work in the same way. Corporately owned processes are primarily templates for group use. It is expected and desirable that groups will customize them or develop their own standard processes. A group's standard processes are intended to be used repeatedly from project to project. They represent a tangible asset that is best managed by the group, which needs this flexibility to maximize the value it delivers to its customers.

A project-level process is the specific sequence of steps that executes the project plan. The process is generally different for each project because of unique customer, environment, or project considerations. However, a project's process should be based as much as is practical on a standard process, one that has already been developed and is in common use by the group. By following the project's process, the project team will produce a result that is predictable in terms of substance, quality, and schedule. By basing that process on a standard one, most of the work of creating the process has already been done.

In a company as large and diverse as EDS, it is a significant challenge to reach the appropriate balance between standard processes that aim for corporatewide consistency and the flexibility to address each customer's unique needs. The EDS approach to methodologies achieves this balance. It allows EDS to leverage the best practices employed by groups scattered throughout the corporation and to have a variety of contributors to the continuous improvement process.

This approach has two main points. First, the responsibility for the processes to be defined is shared between corporate and local organizations. EDS cannot, for example, dictate a single process for systems development that must be used in all business conditions. Instead, groups are expected to customize corporate processes or develop their own standard processes.

The second point is that the corporation recognizes multiple layers of detail in its approach to and definition of methodologies. Multiple methods within the framework and multiple processes within each method may be necessary. This flexibility allows groups to better fulfill their responsibilities.

Process improvement has been shown to reduce errors, increase productivity, and provide a consistency of results, leading to customer satisfaction. The entire methodology structure provides constancy of purpose, yet enables an environment that is process driven, corporately supported, locally prescribed, and continuously improved. It allows the benefit of leverage, without the excess cost of misapplied standards. It also delineates the responsibilities of corporate and local organizations regarding the management of standard processes.

Examples of current and future corporate methodology frameworks and their corresponding line of business supported are as follows:

- Systems Life Cycle: Systems Development, Systems Integration, Systems Management
- Business Direction: Consulting
- Business Operations: Consulting
- Resource Management: Consulting, Systems Development, Systems Integration
- Integrated Program Development: Consulting, Systems Development, Systems Integration
- Software Configuration Management: Systems Development, Systems Management
- Legacy System Migration: Systems Development, Systems Management
- Computer-Aided Platform Engineering (CAPE): Systems Integration
- Product Life Cycle: Systems Management

Examples of supporting processes used in more than one methodology framework are as follows:

- Business Continuity: Systems Development, Systems Integration, Systems Management
- Project Management: Entire Continuum
- Metrics: Entire Continuum
- Requirements Determination: Entire Continuum
- iQ: Entire Continuum
- Change Implementation: Entire Continuum

Hewlett-Packard

1. Changes in the Software Building Process (Overall)

The driving force behind changes in the systems building process at HP is to reduce the time required to deliver projects while improving the overall quality of the finished product. HP's area of emphasis is on the management process for building systems, not on tools. They are placing emphasis on the re-engineering aspects of the business process, and simultaneously trying to develop a methodology which will apply a rigorous discipline to the resulting systems implementation.

They do believe that tools are important. However, they believe that 90% of the errors occurred at the front end, where few tools are of much use without an overall re-engineering of the business process.

They believe that their emphasis on re-engineering focusing on downsizing, organizational changes, process re-design within reasonable cost containment guidelines of the client, is the primary thrust of their rethinking of the systems development process.

Reduced cycle time on the completion of business systems re-engineering project, including the generation of finished applications systems - 25%. They would not give a specific example.

HP is trying to define quality as "fit to business need" within the client's anticipated time frame. Their position is that counting bugs in the code is secondary.

HP has not made much progress on reusability. Reusability per se is not a goal. Studies that they've conducted indicate that there is huge waste in the current process which could be significantly reduced if a set of standards could be developed along with a process to better manage code; identify and select modules to be developed and maintained as "reusable", etc. However, they feel that the process is a long way off.

The main problem is that in order to generate reusable modules, the development process must be integrated or contain components and standards which insure reusability. They think that most firms are farther behind in this area than they are in CASE. (In HP's view, most firms aren't very far along in the effective integration of CASE into the development process.)

Very little is being done to obtain any more meaningful quantification of benefits than was done in the past. This process becomes even more complex when significant business re-engineering is involved. In some instances various components could be readily quantified, but that inmost instances where business investment on the basis of much softer criteria.

2. Soft Benefits

HP's experience has been that the two dominant soft benefits that firms are using to justify re-systemization (and business re-engineering) efforts are:

- Improved customer satisfaction; and
- Reduced cycle time on core business transactions

Certain of HP's customers have taken to trying to measure both of these factors; but it's still difficult to relate the measures to the bottom line performance of the firm.

3. Relative Contribution of Tools and Processes

HP believes that the biggest contributions are coming through improvements in the process and people management aspects of the systems building process. CASE tools and other automated approaches are important but secondary. They have adopted an approach which pairs top-flight project managers with tailored teams for specific implementations. Project managers are responsible for projects only. Team members are managed in the personnel and development sense by other managers within their home organizations.

- They have installed new processes that permit feedback between project managers and those managers in the firm who are responsible for the development and administration of individuals within the organization.
- They have installed a special program to continually improve even seasoned project managers skills.
- They are using some incentive programs for project performance and quality on an experimental basis, but these programs are immature in that they tend to reward heroic fire fighting efforts...rather than solid fire prevention on a day to day basis. Incentives (both team and individual will probably play a stronger role in the future.

4. Reusability

Very little is being done by HP in the reusability concept as applied to custom code development. However, a significant amount of work is going into refining process which will insure reusability of some significant portion of the code that's developed for applications or systems software products.

What is being done in this area is proprietary to HP and its products. They are still a long way from the point where the generation of "universal objects", etc. will be commonplace

5. Difficulties in Addressing Client/Server

HP has been working at the C/S model in an open systems environment for some number of years. They are very comfortable with it at this point, and have a formally documented approach for allocating functions and data between clients and servers. (As is the case with reusability, the model is proprietary to HP and might not be applicable to other platform architectures.) The model is supposed to be rigorous and has significantly improved the quality and efficiency with which new applications can be developing using C/S.

They still needed to improve the overall process at the design level. The area that needs the greatest improvement is a consistent approach to linking redesigned business processes with functional specifications for C/S systems. They are working on this.

- They have just dropped KnowledgeWare in favor of IEF as their primary CASE vendor.
- Anticipate that IEF organization will work jointly with them on tailoring CASE tools to meet HP's needs both at the design and implementation levels.

A formal model for the C/S design process has significantly improved the quality and reduced the time required to systematically test new code. Estimates that they are saving about 20-25% in this area, and are certain that the quality of the code is higher.

6. Is More Than One Process Model to Improve Systems Building?

HP is definitely trying multiple approaches to improving the systems building process, but the process is not seen internally as very well coordinated. Individual initiatives appear to pop up, and get piloted. Informal communications between people involved in these efforts are the primary method of coordination. However, enough information gets shared to insure that some "best of breed" approaches are emerging.

There is a group manager whose responsibility it is to gather, digest and redistribute information on the new techniques and approaches that emerge. Overall, they have a medium to low degree of control over the management of change.

7. Skill Needs and Distribution

Technical skills don't seem to be a problem for HP. The company's culture supports investing in employees, particularly in technical education. Project management is the primary area where they see a significant shortage. And, there are no magic wands to wave that eliminate the need for the "experience" factor in project management. They are trying an "understudy" approach which pairs less experienced project managers with more experienced ones to hopefully obtain knowledge transfer.

8. System Testing Process

Adoption of the C/S model has significantly changed (and improved) the testing process. (See the answer to question 5.) Since the modes provides for a limited set of structured communications between client and server using standard protocols, it's possible to develop test packages to automate the testing of each units functionality separately and in most instances simultaneously. This has lowered testing costs and cycle times significantly.

9. Particular Innovations

They believe that there is significant variation between major firms, but that probably all leading firms have hot spots where advances are being made. Several user companies and two vendors were cited.

- Dupont - Significant work in "rapid" prototyping for C/S applications
- Farmers' Insurance - Advanced architecture for C/S transaction management
- Microsoft - Use of advanced tools and concepts (object oriented coding, etc.) to automate the systems building process. Reusability at applications product level.
- HP - C/S applications model and transaction management
- Andersen - Re-engineering process and (probably) reusable applications modules.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

The following is HP's educated guess:

- In addition to whatever investment cost their might be in designing a new process or technique, the first time the process is put into use on an actual systems effort, there is probably a twenty percent penalty in development cost.
- On the second attempt the penalty probably drops to 10% and on future efforts gains in productivity can be made in the order of 10-20% per effort.

To pull together HP's C/S model probably cost between $500K and $1M. This doesn't count the many hours that were invested in developing stand alone components which were used in the model. However, benefits are there from the developers viewpoint.

After two years of working with the model in actual development efforts, they are probably delivering systems at 80% of the cost for these same systems developed using conventional methods.

11. Success/Failures

The C/S development model has been Hp's biggest success. On the failure side, early on they had an occasional disaster in terms of actually using the model to do a client implementation.

12. Other Organizations to Watch

HP is probably ahead of the average firm in rethinking and actually implementing changes to the software building process. They believe they were probably far from the head of the pack. They don't think that there was any one firm who was the leader in all aspects of the process.

13. Charging for Higher Value Software Assets

HP doesn't feel that the asset value of individual custom designed systems will improve significantly as the direct result of improvements to the process; but does believe that the process will create new kinds of assets.

The business process models developed as a result of re-engineering can and (in a few instances) are being sold or brokered. Repositories of basic business processes will one day be sold. Using these models through a medium such as IEF, other firms could tailor both the processes and the code to meet their particular business needs. One example cited was an airline used TI to broker its IEF model for frequent flyer operations to another airline serving a non-competitive market.

IBM Corporation

1. Changes in the Software Building Process (Overall)

There are changes being made by groups in a number of areas including professional services, external consulting, SI projects, ISSC and internal software groups. IBM's interests include business systems planning, project management, methodologies and tools. Separate activities are also being carried out with business partners.

Internal professional services and software groups are under pressure from users to speed up the development process or time to develop an application system. The reduction of development time by 50 to 100% is mentioned by users as an objective although any meaningful reduction in the overall process (e.g., 20 - 33%) would be well received. IBM recognizes these objectives, but does not have a specific target. However, IBM recognizes that changes of this magnitude will require consideration of all aspects of the process from concepts of business analysis and re-engineering to project management and code generation.

Users also want to reduce the cost of developing software for a system. The reductions being sought by users are at least 30 to 40% and as much as 100% Although there is not any official target, IBM research in Endicott, NY has talked about helping users achieve a reduction of 25 to 30%.

Another objective is to allow the time and cost required to modify software to be reduced.

A different type of objective is to develop software with a higher level of quality. In addition to programs that can be more easily modified and understood, two concepts were mentioned in regard to this. Users have stated that they are interested in systems building that would automatically incorporate various types of input data validation and other cross footing that is suitable for the type of application being developed. The second concept involves enriching the development process or adding tools which will check out the logic of software during the process and leave a structure in place for acceptance testing and testing of modifications at a later date. Auditing should also be facilitated.

Many ideas about quality are prevalent. One recommendation mentioned by users is to ensure that developed software should not be fragile or become fragile when normal types of changes are incorporated or added. Users also feel that procedures or tools should guaranty that software modules can be reused or used as an object(s).

2. Soft Benefits

Software should facilitate efforts to communicate as well as to analyze what is going on in business processes. There should be a view(s) of software that enable users to review systems and give them a feeling of satisfaction with the software.

3. Relative Contribution of Tools and Processes

CASE won't achieve all the objectives that are being discussed for software but CASE tools can aid by enabling software to be analyzed, modified and generated more easily for a number of applications. It was noted that certain applications that require high levels of transaction processing might require attention from personnel that were experts in the alternatives and tools available to meet tasks. One project mentioned in this regard involved an RS/6000 system that AIC worked on in conjunction with the research staff in Austin.

An expert on GUI has been hired by IBM in Endicott, NY to develop special tools to support and guide developers in implementing certain types of workstation systems faster and with more quality.

In regard to the use of tools, it is felt that AI and particularly expert systems, as well as other capabilities should be able to be incorporated as objects in the software development process.

It is felt that the process of systems development must address a much broader set of needs than tools can address. For instance, it must address changing work structures such as work groups and other changes in work organization or people management.

Ideas for the application of manufacturing techniques to software building are also being considered, tracked and experimented with although it is felt that it may be difficult to achieve other objectives such as improvements in communication as mentioned above together with software manufacturing.

4. Reusability

Reusability was mentioned several times as an objective in relation to the goals stated above. The objective was stated as creating software that can be reused and/or facilitate the reuse of other software whenever feasible.

Ideas have been tried out internally by IBM in Boca Raton, Austin, and other locations to reuse software during systems software development.

Research projects have also been carried out to test reusability and other concepts and to track user and vendor work with reusability.

5. Difficulties in Addressing Client/Server

Difficulties involved in separating processes or splitting the management of data have not been fully addressed. IBM thinks that these problems have often led to the use of larger capabilities, AS/400s rather than work stations, in many client/server systems.

6. Is More Than One Process Model to Improve Systems Building?

IBM has a number of process models available to improve systems building and is constantly in contact with other vendors to make arrangements to share and try out their ideas (e.g., recent agreement with HP on client/server development products). They have a low degree of control over the management of change. Many offices (but not all) are supportive of change due to this. IBM contacts in professional services assignments feel that this emphasizes that the most important element in the improvement of systems building is the experience and knowledge of industries and application systems.

IBM points out that some large corporations such as several of the top banks are trying to use more than one process model for improving systems building. This has occurred chiefly due to unplanned pressures by users for different alternatives for systems building. There may not be an adequate means of managing or coordinating change in these situations even if IS tries to coordinate or manage change in situations where more than one model is operative.

Some areas of organizations are more supportive of change, particularly users who are under pressure to achieve business goals.

7. Skill Needs and Distribution

Skills are still in short supply and distributed unevenly. Hiring restrictions and staff reductions have made it difficult to get the right people. The IBM culture still places great emphasis on training. Proactive steps to improve the situation include arrangements with vendors that can aid with or participate in improvement in software building.

8. System Testing Process

Users are being brought in to systems testing to a greater extent. New types of tools or systems review are being sought to improve the ability of users to test systems. Where vendors are involved in the supply of or development of software products, more accountability is being placed upon them during systems testing.

9. Particular Innovations

Innovation of various types is being tried by corporations in contact with IBM including the development of specialized tools internally or by vendors, the generation of objects in C++ code that can be used to add necessary functions to a number of systems, the use of specialized CASE approaches, greater reliance on vendor application software products and experimentation with new types of languages. One effort that IBM is engage in with a user involves the use of specialized GUI that aid a developer to combine objects. Use of the GUI will guide (and force) the developer to include various types of controls and aids that will facilitate testing and operation of the resulting system.

IBM is also conducting research on ideas about GUI that use symbols other than icons. One of these concerns the use of symbols that would aid a person involved in a work group structure.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

The cost of making improvements can not be tracked on an overall basis in IBM and is not being tracked in most organizations. IBM is willing to make improvements to obtain near term business or to improve capabilities over a longer term. It is felt that many short term efforts should be undertaken in order to ensure that IBM will be aware of the more outstanding improvements.

11. Success/Failures

Failures or the inability to achieve total objectives has been notable in some CASE projects as well as projects run by IBM that were supposed to improve software building. A major lesson that was learned was that the business objectives and processes are sometimes neglected or not reviewed sufficiently and rethought during projects that automate systems building.

12. Other Organizations to Watch

The vendor mentioned most often as one who is or may be achieving such improvements is Microsoft. They have been building and rebuilding software objects so that they can be joined together more easily. Visible Basic has been improved and re-coded in C++ to provide users the means of reviewing and adding to or modifying objects in a more easy fashion. The Next operating system of Jobs was felt to be a good step in the direction of improved development, but is not spoken of lately. Products such as Powerbuilder and the Insynch Passport system that generates C code are also being reviewed in addition to the products of TI, KnowledgeWare, Bachman and others whose products are under constant review.

13. Charging for Higher Value Software Assets

Although it is believed that higher value software assets will be developed, there are unanswered questions about the ability of a vendor to achieve sufficient recognition for the software assets put in place so that a premium can be charged on a near term basis.

Microsoft

1. Changes in the Software Building Process (Overall)

The major change is the movement toward larger objects. This enhances reusability. They have customers who will treat the "entire" of Excel as an object in an application. Objects were considered building blocks a long time ago, but now the ability to embed whole applications environments into a system is what is happening now.

The second biggest change is that within two years there will be a new channel in the market place that will cause market shifts -- selling "solution based objects". At first this will only happen within a given software architecture ... Microsoft, Oracle, etc. However, as standards continue to narrow down (e.g., DDE -- Dynamic Data Exchange), these solution objects will be architecturally open; the openness will be limited to a small number of popular architectures. Entire shrink-wrapped applications will be capable of being embedded into solution packages.

The third biggest change is that there will be new methodologies to facilitate the use of these objects.

The benefits fall into the following categories:

Productivity/Shorter Time

Microsoft believes that the main criteria is the time to complete development in the case of information systems and the time to market for software object and package developers. Microsoft estimates that the applications built through some sort of object methodology with reasonable tools can be completed four times faster than through the use of traditional methods.

Quality

Fully tested larger objects means that there is less chance for code test failures.

Graphical interfaces means high levels of customer satisfaction. Embedded help and AI techniques will ultimately reduce the cost of support while making an order of magnitude improvement in quality.

Reusability

They don't know exactly, but believe that it will be orders of magnitude increase shortly. They have been working with a specific customer where credit approval is used in several applications. The original module was built in ACCESS. Microsoft says that they are able to drag and drop the credit module from one application to another and through DDE link the appropriate variables between the object and the new application.

Their outside estimate is that there should be 50% reusability within the more common cross-industry functions within a few (three) years. This high level "solution" object concept coupled with better object linking and C/S design tools could generate even higher levels of reusability.

Quantification of Benefits

Microsoft sees their corporate clients increasingly quantifying benefits, but each in their own way. Microsoft is taking part in the metrics standards group. In spite of this involvement, Microsoft still believes that time to completion will still be the most important measurement, coupled in some way with a measurement of the ease of modification and maintenance.

Microsoft has done some research on 25,000 applications that have been done with their products and they say that in at least 50% of these cases a quantification process was used as part of the justification.

2. Soft Benefits

Microsoft cited a situation where a client of theirs was able to double the number of transactions processed per resource with a
100% improvement in measured customer satisfaction. They also
cited an a telecommunications company which believes that it is saving $50,000/day through a reengineering of the process for allocating "transient" (movable) cells for peak calling periods using C/S technology. To Microsoft, these are examples of how this type of technology can improve communications within an organization.

3. Relative Contribution of Tools and Processes

Microsoft believes that tools are ahead of processes in terms of contributing to success in improvements in the systems implementation process. Even though Windows NT is just being released, there are already 70 tools which have been tested and will be released soon. Whether they are all good tools is another question. They do not see process(es) moving so quickly.

From an organizational viewpoint, they believe that multi-disciplinary teams are important. This is the only change in the process that they see has made a difference.

In terms of methodologies, they believe that top down methodologies are dead, including FOUNDATION and other approaches which were designed to constrain the design to available architectures. While this class of methodologies can be adapted to a degree to current design and implementation architectures, many of the fundamental propositions upon which they are based are no longer true.

Microsoft is developing a methodology designed for developing solutions for C/S environments. It will be full bore, i.e., it will start with business re-engineering and end up with code. They expect to sell this approach directly to user organizations and/or market it through professional service firms.

4. Reusability

See point 1 above.

5. Difficulties in Addressing Client/Server

Microsoft believes that there will not ever be a perfect C/S model, because C/S technology needs to be adaptable to the solution, not vice-versa. The tools in place are increasingly sophisticated -- how to use them effectively is another matter. The problem is educating users and providing design processes which deal with new paradigms of information processing, including graphical and multi-media communication interfaces. Things are moving so rapidly that the problem will soon be selecting the appropriate model for the desired solution, not whether there are enough models around.

6. Is More Than One Process Model to Improve Systems Building?

Yes, there will be more than one approach in the short term because there are not yet clear cut winners. They are testing the waters. Ultimately they want one approach.

Change is managed or coordinated relatively tightly (4 on a scale of 1 to 5, where 5 = a high level of management control)

7. Skill Needs and Distribution

Yes, there is a severe shortage of skills overall in the firms that Microsoft works with; Microsoft can't supply all the skills themselves, so is dependent partly on its users and customers. The former business analyst must be transformed into someone who can support re-engineering or processes and build prototypes of applications using high level tools. More technical staff will become module or "component" builders.

The problem is significant because the formal processes, standards, tools and educational methods are falling into place to make these new skill paradigms happen.

Basically, Microsoft feels that because there is a high need for transformation and evolving support tools/processes, that the market will make this adjustment in skills occur.

8. System Testing Process

Integrated testing used to be the final step in a new system. Under the new C/S model, testing will occur simultaneously across the entire development process. Individual modules (objects) will be bug-proof. Code will integrate modules into solutions and will increasingly use high level meta languages. This will reduce the risk of "hard" coding failures.

The net effect is that time and costs will become significantly less over time.

9. Particular Innovations

[Several case studies are promised, but have not yet been received.]

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

The costs are significant, including technology and training. However, many of the costs are being absorbed as part of the development effort so are not really being measured.

The data that Microsoft has indicates that identifiable maintenance costs for C/S systems will probably be 50% of the original systems cost over a five year period. This compares to 70% for systems implemented under traditional architectures.

They have had some customers who feel that they got payoffs from applications implementations within weeks or months of implementation. Many of these applications were created within comparable timeframes.

11. Success/Failures

They would not comment on Microsoft failures except to say that these have been covered in the press, but were not fatal.

In customer implementations, the biggest "failures" have been in the management of user expectations. A second factor is the selection of "wrong" tools -- the tools may have been ok at the beginning but either new tools become available and/or there is a truer appreciation of what tools were needed. Both these factors will cause confusion in the market and increase the difficulty of measuring success.

12. Other Organizations to Watch

They feel that Lotus has done good work in the applications and applications template areas (referring, essentially, to NOTES), but that Lotus had ceased being a leader.

They think Borland is poised to make large contributions. Borland doesn't have the leverage of Microsoft, but have done a lot of good thinking.

In terms of the "traditional" players, they think EDS is way ahead in its thinking in terms of utilizing C/S technology. DEC may be there soon also. SHL is also doing some interesting things in C/S applications integration.

13. Charging for Higher Value Software Assets

Software asset value will go up and up. The more software becomes integrated with individual business functions, the more it will be treated like "plant and facilities". Can anyone measure it? No! But Microsoft doesn't doubt that it's there. Over time there may be some way to measure it.

Oracle

1. Changes in the Software Building Process (Overall)

The principal changes taking place in the process are driven by three fundamental driving forces.

- A migration from IS to the end-user as the leading player in the development process.
- The maturation of a new life cycle based on rapid prototyping, utilizing increasingly sophisticated tool sets.
- A growing demand for rapid turnaround and responsiveness for changes to whatever system is in place.

Both downsizing from the mainframe and upsizing from standalone departmental systems to departmental work group systems is occurring simultaneously. In either case, the traditional life cycle that was used for mainframe development is or will be replaced by the cycle described below.

- Rapid prototyping of a systems basic functionality. (Joint effort of specialists - either vendors or IS - and end-user experts.) The prototype then goes into limited production.
- Refinement and rollout of the full blown system. (Joint effort of specialists and end-user experts, and based on feedback from operating the prototype in production mode.)
- Integration of the refined system with related systems. (Primarily a specialist implementation activity based on requirements identified while operating the refined system in production.)

In other words the traditional concept of a formal requirements phase is replaced by three less formal requirements definition phases. In fact each phase of the traditional life cycle has a three counterparts in the new cycle; design, development, testing, etc.

Oracle sees itself as being involved in this process in several ways:

- As an innovative tool supplier
- As a teacher/coach
- As a high level consultant and/or implementor
- As a developer of applications software (for internal use or, perhaps, in partnership with a client)
- As a supplier of flexible applications (where Oracle may again become involved as a teacher, consultant or implementor)

These are complex roles and Oracle is the first to admit that everything isn't sorted out yet.

Meeting the real business requirements of the end-user, and shortening the time from inception to delivery in order to capture user benefits early. This fundamentally comes down to:

- Reduced training time
- Shortened development cycle to first usable version of the system
- User self-sufficiency in terms of on-going modifications, etc.

Oracle does not believe that there is likely to be a reduction in systems development costs (at least in terms of reduced manpower) over time. The more likely scenario is that implementations will have four to five times the functional capability for the same cost.

Oracle believes that the creation of new interactive applications using their redefined life cycle and tools can be accomplished in 70% of the time it would take to achieve the same functionality using conventional CASE and methods. For batch based applications or the generation of new reports, etc., they believe the number is 25%. (See attached diagram showing how newer approaches to system building generate benefits.)

Quality is customer satisfaction with the provided functionality, and "0 defect code".

In a homogeneous environment (Oracle Platform) they estimate that a fair amount of the code can be re-used. But this is not an objective unto itself. In heterogeneous environments (say using DB2 along with Oracle), less will be achieved in terms of reusability.

Oracle believes that the industry is a long way off from obtaining reusability on any major scale through the development of "objects". Probably the highest level of reusability comes through the creation of user "customizable" applications software products.

Very little quantification of benefits has been done. Most end-user projects (which are an increasing proportion of the total) seldom undergo a formal justification or subsequent audit process.

2. Soft Benefits

As firms begin to leverage the concept of information distribution and collection utilizing the interactive capabilities of modern technology significant benefits will be realized in terms of better communications both deferred and direct. This will include interaction between individuals in a firm as well as between an individual and the corporate repository of knowledge. Applying these same systems design concepts to interactions with customers, etc. will bring another whole set of benefits.

The typical knowledge worker spends between 10-15% of his/her time today interacting with paper. As the level of integration of systems goes up and the ability to interact on an as needed basis with the corporate repository grows, knowledge workers will spend probably 20-25% of their time working with information electronically.

In Oracle's view we're just seeing the tip of the iceberg in terms of understanding, articulating and realizing some of the significant benefits that can be achieved through the migration to interactive systems at all levels.

3. Relative Contribution of Tools and Processes

New tools and changes in processes are probably making an equal contribution; and they continue to evolve in an iterative manner. The methodology that Oracle endorses of rapid prototyping, refinement and integration has stimulated the need for new functionality in the tool set. And, the capabilities that are subsequently generated in the tool set provide opportunities for further refinement in the design and implementation processes.

CASE tools are either obsolete or essential. If they are integrated with rapid prototyping methodology and deal with the concept of distributed architecture, they are essential. Tools that "assume" the kinds of structured methodologies of the 70's and 80's are probably obsolete because they don't integrate with current methodologies or architectures.

AI, like many other buzz words (including CASE), has gotten a bad name. However, if you look around, you'll see that expert rule based systems are imbedded in everything from spreadsheets and word processors to systems development tools. The use of these technologies to support the systems building process will continue to grow, and will result in both reduced development time and higher quality code.

The biggest single change on the people and organization side was the migration of development responsibility from IS to the end-user. Distributed IS has changed the process. The result is more teamed approaches which hopefully combine the required technological expertise with user knowledge of business process.

In terms of management techniques, people are still going to be people and managers will still vary in quality. However, the teaming of users and technical personnel in the development process will and should continue to reduce the complexities of management by improving communications. New tools and processes will continue to reduce the number of people required to accomplish an implementation (and therefore the complexity).

Oracle believes there is some fallacy in the notion of "manufacturing" code. Manufacturing implies making a large quantity of the same (or essentially the same) thing. Building applications involves using common processes to generate unique systems. The thing that they have in common is the use of consistent (but obviously different) processes... little else.

4. Reusability

(See Question #1)

From Oracle's point of view, reusability is an objective. However, a lot of effort will need to be placed into the development of standards, and the consistent use of sophisticated tools before this will be accomplished through "object" libraries. In the meantime, Oracle targets on being able to reuse higher level application definitions to "regenerate" code. Modifications to the definitions at the high end allow customization; and the use of tools allows new custom code to be generated at minimal cost.

There may not be much progress was likely to get made in generating reusable modules at the user interface, since this is an area where systems providing similar functionality were liable to vary significantly based on personal and organizational preferences.

5. Difficulties in Addressing Client/Server

Oracle believes that the C/S model is incomplete. When utilizing a consistent platform on any given set of applications (the Oracle platform) they have a consistent way to model the distribution of processing and data. It's pretty straightforward. Shared functions and data go to the server, and unique functions and data go to the client. However, when working with heterogeneous platforms (in particular data bases), the ability to follow the model is frequently inhibited by differences in functionality in hardware and software.

Over time, the migration to more open systems and standard interfaces will likely increase the level of consistency that can be achieved in allocating data and processing in the C/S environment. In any event, its not a serious enough problem that it is likely to inhibit the ongoing emergence of C/S as the primary applications platform.

6. Is More Than One Process Model to Improve Systems Building?

Yes, Oracle is certainly trying more than one model. Two examples have already been cited; the rapid prototyping process and the creation of "customizable" applications packages. More are likely to be tried in the future.

In terms of how the process is being managed... at Oracle it's ad hoc. Overall, they have a low degree of control over the management of change.

7. Skill Needs and Distribution

Oracle, being a technology company, makes heavy investments in training, and provides numerous incentives for self-development. On-going training and development is just part of the culture. So fundamentally, Oracle doesn't feel that it's facing a any significant problem in finding or developing the skills that it needs.

In general, Oracle sees the distribution of IS skills to user organizations is a significant change, and will continue to happen throughout the decade. In terms of skills bottlenecks, Oracle believes that these are largely transient effects.

8. System Testing Process

Testing is becoming an on-going continuous process. The objective is no longer just to meet a written specification, but to insure that the customer is satisfied with the cost, functional performance and time to delivery of the system. In a sense this increases the complexity of the entire testing process, and places the "pass fail" judgment directly to the user.

In terms of actually testing finished code, major tests are built in at each of the three major phases of the Oracle process (rapid prototyping, refinement, integration).

The process of regression testing of software is growing more complex. As more and more functionality is included, the possibilities that need to be tested multiply exponentially. Some automated processes have been put in place, but manual testing of individual products is still required.

9. Particular Innovations

- More and more integration of applications systems and data
- Installing methodologies which permit the continuous refinement of systems
- Providing tools and methodologies which allow users to achieve high levels of self-sufficiency in terms of managing their own systems environment.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

Unfortunately this is something that isn't measured. Partly because what is being done is intertwined with so many other activities, it's not clear that even a heavy investment in the measurement process would produce a result of significant value. However, major investments in new methodologies and tools were looked upon by users as long term investments. Only when the first applications came rolling out under the new approach would people begin to realize the potential savings or pay back.

11. Success/Failures

It's no secret that Oracle's biggest failure was in its initial approach to the applications software market. The products (originally developed for internal use) were low in quality and functionality, support was low quality, and sales commitments exceeded the deliverable capability. This problem is behind them now, and caused a major rethinking about the entire applications building process which forced Oracle into its new direction - with benefits all around.

12. Other Organizations to Watch

Some significant improvements are being made in the pharmaceutical industry. (Examples which confidential.) There are similar developments in the chemical where the adoption of new techniques and approaches to systems is allowing high levels of integration improving overall productivity to the firm.

13. Charging for Higher Value Software Assets

The actual asset value of software will probably go down as it becomes easier to build and is replaced more quickly. However, the value of the "information asset" will likely rise significantly as newer applications and software systems permit its distribution, collection and utilization in ways that weren't previously possible. Whether anyone can measure these values is debatable.

Texas Instruments

1. Changes in the Software Building Process (Overall)

The major activity at TI is a total restructuring of the development process across all organizations. The corporate organization is managing this change in process. They've have been working on this for about one year. They are building their process on the Software Engineering Institute (Carnegie) model. Three key components to the model that they're emphasizing; (1) Dynamic (living) specification systems, (2) New quality assurance processes, (3) Configuration control and management.

From a corporate viewpoint they're interested in controlling an overall template for the development process. Each division will be allowed (within guidelines) to modify the process according to their specific needs.

- What benefits:
  - Productivity

    They have no history against which to measure productivity improvement. They are currently populating a data base which is capturing their results with the new approach. However, TI believes from the top down that a consistent adoption of the new methodology will yield consider savings. (The new methodology contains more sophisticated metrics which focus on user requirements in addition to productivity from the viewpoint of system development.)

  - Shorter time

    Have experienced decreases over originally anticipated schedules. Less overruns, but can't document improvements based on historical experience with comparable systems.

  - Quality

    From the IS viewpoint, there is considerable improvement. They've installed some systems which went in "bug-less", and seem to require considerably less maintenance than most. Although TI hasn't figured out how to measure it, they believe that there will be a two tiered measurement system: Technical quality and compliance with user needs and expectations.

  - Reusability (how much?)

    Have not had time to focus on this area. It's a longer term objective, but have done practically nothing to formalize an approach to reusability. They feel it is pointless to address this issue until they had gotten a common process in place. Furthermore, most of the work that they are involved in internally is strictly custom. There are probably not very many applications modules that are reusable.

- Quantification of benefits?

  In terms of the cost of implementing the new systems building process, a superficial cost justification was done base on other firms' data who have adopted the SEI approach. However, top management is willing to foot the bill, just because it makes sense. (Their initial successes have provided additional reinforcement to this concept.) She also said that from the point of view of internally developed applications, line management is pretty serious about justification from the top to the bottom.

2. Soft Benefits

Probably the real payoff here is in quality of the products and services that TI can deliver. Very little measure, but many claims; and recently some "user" sponsored endorsements. The biggest qualitative benefits have been achieved due to the improved communications that occurs between developer and users in the systems development process. They have examples of five projects which have been delivered under the new process where the user client expressed total satisfaction with the delivered system, and in fact, had no immediate requests for changes or refinements.

3. Relative Contribution of Tools and Processes

The process must be in place first. They have deliberately put manual systems into place first. Once there was experience with the process, they have been able to develop specifications for what kinds of tools that they need. They have since been adding the tools that the feel will give them maximum leverage.

They do use IEF, and they are primarily a large mainframe shop, but have been preparing for the kinds of processes and tools that they will need for the inevitable movement toward distributed systems.

Recently they have installed a tool to assist them with project scheduling, and are examining alternatives for C/S environments.

4. Reusability

(See Question #1)

5. Difficulties in Addressing Client/Server

The demand for C/S has been growing at TI. Many implementations have already been done. However, the model is definitely incomplete. They have 12 people dedicated to developing the methodology and tool sets that would be required to place C/S development in the context of their overall project development and management scheme. They are hoping to be in front of this situation, but there is no clear cut approach which seems to be dominating.

6. Is More Than One Process Model to Improve Systems Building?

TI's corporate strategy is to have a single general approach. TI feels that this is working. In essence they require compliance with the overall model, but are flexible as to how individual divisions tune it and select tools that meet their particular needs. The process isn't totally top down. There is an organization that functions as a "switch" for new ideas regarding innovations that might percolate up through the divisional organizations.

7. Skill Needs and Distribution

They are finding that there is a strong need for people with specific skills. Specific skills which appear to be in short supply are business engineering, quality assurance, and project management. TI feels it is able to overcome the demand for new types of technical skills through training and hiring. They find themselves with a very limited set of people who have the non-technical experience to adapt to new methodologies. These people are harder to find, and training is not nearly as effective as on the technical side.

To overcome this problem they have developed a series of "centers of expertise". They use teaming to staff projects. They used to expect their "analysts" to be able to do everything from business analysis to systems test. They are now attempting to maximize the utilization of the scarce skills through teaming.

8. System Testing Process

Using a more iterative approach which is consistent with the tighter coupling of users and developers in the new systems building life cycle. TI feels that as far as the code testing process, they've had an advanced system for some time. Nevertheless several initiatives are going forward to insure that whatever improvements can be made in testing, are.

Essentially, they have an "early" inspection process for testing. Same as most other modular approaches.

9. Particular Innovations

TI considers itself very innovative in their quality assurance process. Instead of after the fact, their quality assurance process is focused on identifying failure risks up front, and clearing the way for success. The process is iterative.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

TI thinks that changes in the systems building the are definitely a long term investment. They consider their overall program a 10 year initiative. This is particularly true in terms of dealing mainframe and legacy systems. However, they believe that they are already seeing benefits in terms of higher quality, shorter completion times and reduced maintenance costs. Systems developed under their new methodology appear to be yielding these benefits consistently and immediately. Their preliminary measurements under the new system showed reduction in maintenance and enhancement costs of 40%.

11. Success/Failures

Five major systems that have been built under the new systems methodology and are seen as successes. The corporation seems to be adopting the strategy rapidly and only with pockets of resistance.

TI feels that it is way ahead in business process engineering. They are already selling approach to certain business processes on a one-on-one basis. They anticipate that they will continue along these lines.

12. Other Organizations to Watch

In their industry TI believes that Hughes, Motorola and Raytheon have made significant progress (beyond TI's) in improvements to the systems building process. However, TI is ahead on business process re-engineering.

13. Charging for Higher Value Software Assets

The asset value of software can only be measured in terms of business value. Very little progress has been made with these kinds of metrics. The fact that they are able to sell on the outside some of the business re-engineering models shows that it is possible to get a measure.

Xerox PARC/XSOFT

Background: XSOFT acquired by the Xerox PARC operation to partly to be a conduit for the commercialization of software products or concepts that it has developed. In many instances they, XSOFT, outsource code development to the PARC.

1. Changes in the Software Building Process (Overall)

The principal changes are taking place in the process and are driven by three fundamental driving forces.

- The use of higher level languages in the creation of products (C++)
- The switch to object orientation (even if on a proprietary basis)
- Innovations in the systems/products building process itself.

Changes in the process include:

- More structured disciplines on the front end. Less "adhocticity" (their word)
- Heavy emphasis on joint design (user/designer)
- Heavy emphasis on quality
- They estimates 60% of their efforts are spent in the design phase (to prototype), 40% on actual implementation of code.

They have reduced time required by 20% to migrate existing products to new platforms: proprietary RISC to open RISC to Intel, etc. The first roll over to a new architecture took 1.5 years, the second, about 1 year, and the most recent just 9 months.

Quality improvements are along several dimensions,

- Reduced defects - 30-50% less bugs using newer approaches and tools.
- Reduced severity of defects - They use a scale for calibrating the level of defects from the design level to code errors. They wouldn't quote a number for this measurement but implied that they actually find very few code errors or design errors any more (the high and the low ends of the scale). The most severe errors they were turning up were due to anomalies in the operating systems for various platforms they support for their products.
- Improved customer satisfaction. They make extensive use of customer satisfaction as a measure of quality. They survey every six months and re-survey customers as their use and maturity level with the use of the products grows. They claim that customer satisfaction has gone up, since they have adopted new development approaches. They didn't (or couldn't) cite comparative statistics.

The extent of reusability varies.

- Looking at it from one angle, they get very high levels of reusability on platform migration efforts. Essentially all the core code moves over without modification. The primary areas where they need to custom design and develop come at developing the user interface management and integration at the operating system level.

- In terms of traditional projects where a new system/product is being developed from scratch, they are beginning to reap some of the benefits from their investment in object development. However, most objects are still at a "low" level; i.e. not very complex. XSOFT anticipates that as the number and richness of the objects grow, they will be able to obtain significant levels of reusability. Also this process will be non-linear..."richer objects will spawn even richer ones" (quote).

They don't believe that users do much in the way of quantification of benefits. Generally, PARC's offerings are not specific with regard to applications functions. They believe that when a firm decides to adopt the use of XSOFT products, it is a strategic decision, and most likely justified by functionality required to meet some business critical application decision.

2. Soft Benefits

PARC/XSOFT utilize an executive council made up of key users of their products to help them decide on new directions and additions in functionality that should be made to existing products. This organization frequently cites various kinds of soft benefits as the reasons they feel that new functionality should be added. This organization provided forceful arguments as to why the product line needed to be migrated to other platforms and open operating environments. But in the end PARC and XSOFT made the decision based on their estimation of increased market share for their offerings.

3. Relative Contribution of Tools and Processes

Organizational approach and process are key. The process must maintain a strong linkage between the development team and the end-user through all phases-from conceptual functional design through test and customer support. This is equally true for software products as for traditional applications development. Changes in the process are seen as responsible for 70% of the improvements that they are getting.

From an organizational viewpoint, teams are important. XSOFT uses team reward structures in most development efforts. Teams are populated based on skill requirements, customer communication capabilities, and project management experience. Part of the reward structure is based on the product quality as measured by the customer satisfaction survey process.

4. Reusability

(See question #1)

5. Difficulties in Addressing Client/Server

Yes, the client/server model is incomplete. In particular, most applications function transfer takes place between files, or blocks of code. There needs to be more intelligence between what is transferred between the two peer level environments. In other words the environment needs to be more collaborative. Also, some model for dynamic reallocation of activities between clients and servers depending on the computing environment at any given time.

The fact that the perfect model doesn't exist is not likely to inhibit the continued migration to C/S for new applications. And, once a design decision (based on whatever allocation of functions is decided) is made, the new development approaches using objects, etc. can leverage that design.

6. Is More Than One Process Model to Improve Systems Building?

PARC and XSOFT are pretty will coordinated on managing the evolution for the sytems building process. They term it a "cohesive" approach. They have some "skunk works" groups trying now new concepts, but its's not a competitive ownership issues. These organizations are funded to test out new ideas. The results they achieved are evaluated and changes are made to the core processes to leverage any significant concepts developed by the "skunk works" teams. Change is managed/coordinated very well. (5.0 on a scale of 1 to 5, where 5 = high level of management control.)

7. Skill Needs and Distribution

Yes, there's a shortage of skills; and yes, the distribution is changing. From the actual development viewpoint, PARC/XSOFT is finding that it can do more with less people, but that to achieve the quality levels they strive for, more people time is being invested into customer related activities. The technical skill problem is easier to deal with, in light of the fact that they are primarily a technology company. However, experience is a big factor in terms of developing individuals who can handle the customer end of the process. This area is the biggest bottleneck a the moment.

8. System Testing Process

They don't see much change in process itself. More emphasis is being placed on testing specifications, although this is still more of an art than a science. Over time they expect that 70-80% of the investment in testing will be in this area. Clearly there's considerably more customer involvement throughout the development cycle. But, the process itself is not significantly different.

9. Particular Innovations

There are two areas where PARC/XSOFT sees itself as being particularly innovative.

- From a technology viewpoint - they are investing considerable effort in developing more sophisticated software to apply intelligence to the server. (This issue was discussed earlier in the interview in question number 5.)

- From a business standpoint, they believe that the relationship between PARC and XSOFT is unique and mutually beneficial. PARC places emphasis on innovative techonology concepts, and XSOFT on how they can be converted into good business opportunities.

They feel there were pockets of innovation all over the industry where significant and innovative strides are being made in the way software is designed and built, but didn't think anyone had a monopoly or could be considered the industry leader.

10. Cost of Making Improvements in Systems Building (Funds, People, and Opportunities Foregone)

No way to measure this. The major investment that was made by PARC/XSOFT to move to multiple platforms and open systems was analyzed and costs estimated, but they cannot disclose them. Since then, improvements in both the processes and products have been incremental in nature, and it would be impossible to determine the payback of these incremental changes.

From the point of view of payback, PARC/XSOFT look at the dollars they can pull in from the market. In that sense they feel that they have gotten rapid payback on investments through the migration of their offerings to multiple platforms. It's not clear how long it will take to get significant yield from the "open" strategy.

Although no scientific measure has been done, the changes in the development process which allow product to go to market more quickly with higher quality certainly improve the payback cycle.

11. Success/Failures

Moving their entire product line from proprietary to open in under two years. (Success)

No comments on failures.

12. Other Organizations to Watch

No comments on other companies except to reiterate the concept that there are pockets of leadership all over the industry. From their viewpoint the PARC/XSOFT business relationship is unique and appears to be a good model. This may be an area where they are ahead of some of the rest of the pack.

13. Charging for Higher Value Software Assets

Software asset value will somehow have to be measured in terms of business contribution. The concept of software independent of the entire business process it drives has little or no asset value to the end user. The same software may have a great variety of "asset values", depending on how it's applied.